

NEALT

Northern European Association for
Language Technology

NEALT Proceedings Series No. 45



Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)

May 31–2 June, 2021 Reykjavik, Iceland (Online)

Editors: Simon Dobnik and Lilja Øvrelid

NoDaLiDa 2021

**23rd Nordic Conference on Computational Linguistics
(NoDaLiDa)**

Proceedings of the Conference

May 31–2 June, 2021

Reykjavik, Iceland

Online

©2021 Linköping University Electronic Press

Front-cover photo of the ongoing volcanic eruption in Geldingadalir, near Reykjavík by Kristinn Ingvarsson, University of Iceland.

Published by
Linköping University Electronic Press, Sweden
Linköping Electronic Conference Proceedings, No. 178
NEALT Proceedings Series, No. 45
Indexed in the ACL anthology

ISBN: 978-91-7929-614-8
ISSN: 1650-3686
eISSN: 1650-3740

Sponsors



GRAMMATEK

Message from the General Chair

Welcome to the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa 2021)!

I am a great fan of the NoDaLiDa conference, as a friendly, medium sized conference that offers the opportunity for scientific and social interaction with colleagues from neighbouring countries. When I agreed to serve as the general chair for this years NoDaLiDa I was still relatively optimistic that we could all meet in beautiful Reykjavik, Iceland to enjoy two days of scientific talks, posters and socialising in early June. Unfortunately that turned out to not be possible due to the COVID-19 pandemic. Instead, we are for the first time offering NoDaLiDa as a fully virtual event, free of charge. Even so, I am confident that this years conference will offer the same high-quality program as in previous years and hopefully it can also constitute a meeting place, albeit a digital one, for Northern European NLP researchers in these unusual times.

As in previous editions, the conference features three different types of papers (long, short and demo papers). We received 91 legal submissions, which represents an increase compared to the previous edition of the conference. In total, we accepted 54 papers, which will be presented as 30 oral presentations, 22 posters and 2 demos at the conference. Each paper was reviewed by three experts. We are extremely grateful to the Programme Committee members for their detailed and helpful reviews. Overall, there are 8 oral sessions with talks and two poster sessions organised into themes over the two days, as well as two exciting keynote talks.

I would further like to thank our two great keynote speakers for sharing their work with us: Lucia Specia from Imperial College London will talk about “Disagreement in human evaluation: blame the task not the annotators”. Adina Williams from Facebook AI Research (FAIR) will talk about “For Matters Word Order Little MLM”. Two exciting talks that complement each other well!

As in previous years, the conference will be preceded by three workshops: Translatology in the Digital Age, NLP for Computer-Assisted Language Learning and Sustainable language representations. I want to thank the workshop organisers for complementing the main program and offering opportunities for in-depth scientific interaction on these diverse and exciting topics.

I would like to thank the entire group of people that made NoDaLiDa 2021 possible. First of all, I would like to thank Beata Megyesi for inviting me to take up this exciting (and at times daunting) role and all her valuable input regarding NEALT and previous editions of NoDaLiDa. I am further indebted to Barbara Plank for her encouragement, for the sharing all the great resources from the last NoDaLiDa and willingly answering questions on all aspects of the conference organisation. I want to thank the program chair committee Jurgita Kapočiūtė-Dzikienė, Mark Fishel, Jón Gudnason, Barbara Plank, Yves Scherrer and Sara Stymne, for working hard on putting the program together. I am particularly grateful to Jurgita Kapočiūtė-Dzikienė, Jón Gudnason, Yves Scherrer and Sara Stymne for their great effort in leading the reviewing process and shepherding papers from submission to a final decision. I could not have done this without you! Special thanks go to the workshop chairs, Hans Moen and Ildikó Pilán, who have done an invaluable job with leading the workshop selection and organisation. A big thanks also to Johannes Bjerva for his work as social media chair and Simon Dobnik for leading the publication efforts that led to this volume, as well as the coordination of the workshop proceedings. Thank you! Finally, my ultimate thanks goes to the local organisation committee and team. Thank you, Hráfn Lóftsson, Anton Karl Ingason and Steinþór Steingrímsson. They are the ones who did all the heavy lifting in the switch to a virtual event and did a truly amazing job!

NoDaLiDa 2021 has received financial support from our generous sponsors, which we would also like to thank here: Lingsoft, Tilde, Mideind and Grammatek. Above all, their support made it possible for us to offer this NoDaLiDa free of charge. I hope that this will open the conference up to an even larger audience of NLP researchers in Northern Europe.

Once again, welcome and I hope you will enjoy the conference!

Lilja Øvrelid

Oslo

May, 2021

Message from the Local Organisers

We were very much looking forward meeting you at the beginning of summer in Reykjavik, Iceland, but due to the COVID-19 pandemic we had to move the conference completely online. This has been a challenge for us, given the fact that NoDaLiDa has never been run online before. We looked at various possible implementations, but at the end we selected a combination of Zoom, YouTube, Gather.town and Trello! Hopefully, we have risen to the challenge and we hope that you will enjoy interesting talks, posters, demos and workshops during these three days of NoDaLiDa 2021.

Welcome to NoDaLiDa 2021 online!

Organising Committee

General Chair

Lilja Øvrelid, University of Oslo, Norway

Program Committee

Mark Fishel, University of Tartu, Estonia

Jón Guðnason, Reykjavik University, Iceland

Jurgita Kapočiūtė-Dzikiėnė, Vytautas Magnus University, Lithuania

Barbara Plank, IT University of Copenhagen, Denmark

Yves Scherrer, University of Helsinki, Finland

Sara Stymne, Uppsala University, Sweden

Publication Chair

Simon Dobnik, University of Gothenburg, Sweden

Social Media Chair

Johannes Bjerva, Aalborg University, Copenhagen, Denmark

Workshop Chairs

Hans Moen, University of Turku, Finland

Ildikó Pilán, Norwegian Computing Center, Norway

Workshop Chairs

Hrafn Loftsson, Reykjavik University (chair)

Anton Karl Ingason, University of Iceland (co-chair)

Steinþór Steingrímsson, Arni Magnusson Institute of Icelandic Studies (co-chair)

Invited Speakers

Lucia Specia, Imperial College London

Adina Williams, Facebook AI Research

Reviewers

Yvonne Adesam, University of Gothenburg
Lars Ahrenberg, Linköping University
David Alfter, University of Gothenburg
Tanel Alumäe, Tallinn University of Technology
Krasimir Angelov, University of Gothenburg and Chalmers University of Technology
Rahul Aralikkatte, University of Copenhagen
Jeremy Barnes, University of Oslo
Johannes Bjerva, Aalborg University, Copenhagen
Johanna Björklund, Umeå University
Jari Björne, University of Turku
Bernd Bohnet, Google
Marcel Bollmann, University of Copenhagen
Michal Borsky, Reykjavik University
Gerlof Bouma, University of Gothenburg
Johan Boye, KTH Royal Institute of Technology
Maja Buljan, University of Oslo
Marie Candito, Université Paris 7 and INRIA
Lin Chen, UIC
Daniel Dakota, Uppsala University
Hercules Dalianis, Stockholm University
Dana Dannells, University of Gothenburg
Miryam de Lhoneux, University of Copenhagen
Simon Dobnik, University of Gothenburg
Adam Ek, University of Gothenburg
Anna Esposito, University of Campania “Luigi Vanvitelli”
Filip Ginter, University of Turku
Ana Gonzalez, University of Copenhagen
Johannes Graën, University of Gothenburg
Hugo Hammer, Oslo Metropolitan University
Mareike Hartmann, University of Copenhagen
Daniel Hershcovich, University of Copenhagen
Nora Hollenstein, ETH Zurich
Bjarte Johansen, Equinor ASA
Richard Johansson, University of Gothenburg
Jenna Kanerva, University of Turku
Jussi Karlgren, Spotify
Andre Kåsen, National Library of Norway
Andreas Sjøberg Kirkedal, Interactions LLC
Roman Klinger, University of Stuttgart
Mare Koit, University of Tartu
Dimitrios Kokkinakis, University of Gothenburg
Artur Kulmizev, Uppsala University
Robin Kurtz, KBLab, National Library of Sweden
Andrey Kutuzov, University of Oslo

Veronika Laippala, University of Turku
Krister Lindén, University of Helsinki
Pierre Lison, Norwegian Computing Center
Jan Tore Loenning, University of Oslo
Hrafn Loftsson, Reykjavik University
Juhani Luotolahti, University of Turku
Eydis Magnúsdóttir, Reykjavik University
Hans Moen, University of Turku
Costanza Navarretta, University of Copenhagen
Anna Nikulásdóttir, Grammatek ehf
Mattias Nilsson, Karolinska Institutet
Joakim Nivre, Uppsala University
Farhad Nooralahzadeh, UZH
Pierre Nugues, Lund University
Emily Öhman, University of Helsinki
Fredrik Olsson, RISE
Robert Östling, Stockholm University
Anthi Papadopoulou, University of Oslo
Victor Petrén Bach Hansen, University of Copenhagen
Eva Pettersson, Uppsala University
Ildiko Pílan, Norwegian Computing Center
Mārcis Pinnis, Tilde
Vinit Ravishankar, University of Oslo
Sagnik Ray Choudhury, University of Copenhagen
Matīss Rikters, University of Tokyo
Fabio Rinaldi, University of Zurich
Eiríkur Rögnvaldsson, University of Iceland
Samuel Rönnqvist, University of Turku
Jack Rueter, University of Helsinki
Magnus Sahlgren, RISE AI
Askars Salimbajevs, Tilde
Marina Santini, RISE
Inguna Skadiņa, Tilde and University of Latvia
Maria Skeppstedt, Stockholm University
Karolina Stanczak, University of Copenhagen
Steinþór Steingrímsson, The Árni Magnússon Institute for Icelandic Studies
Umut Sulubacak, University of Helsinki
Torbjørn Svendsen, Norwegian University of Science and Technology
Gongbo Tang, Uppsala University
Samia Touileb, University of Oslo
Francis M. Tyers, Indiana University Bloomington
Martti Vainio, University of Helsinki
Rob van der Goot, University of Groningen
Daniel Varab, Novo Nordisk and IT University of Copenhagen
Erik Velldal, University of Oslo
Martin Volk, University of Zurich

Elena Volodina, University of Gothenburg
Jürgen Wedekind, University of Copenhagen
Anssi Yli-Jyrä, University of Helsinki
Marcos Zampieri, University of Wolverhampton
Niklas Zechner, University of Gothenburg
Heike Zinsmeister, University of Hamburg

Invited Talks

Lucia Specia: Disagreement in human evaluation: blame the task not the annotators.

It is well known that human evaluators are prone to disagreement and that this is a problem for reliability and reproducibility of evaluation experiments. The reasons for disagreement can fall into two broad categories: (1) human evaluator, including under-trained, under-incentivised, lacking expertise, or ill-intended individuals, e.g., cheaters; and (2) task, including ill-definition, poor guidelines, sub-optimal setup, or inherent complexity or subjectivity. While in an ideal evaluation experiment many of these elements will be controlled for, in this talk I will argue that task complexity and subjectivity are much harder issues and that in some cases agreement cannot and should not be expected. I will cover several evaluation experiments on tasks with variable degrees of complexity and subjectivity, discuss their levels of disagreement along with other issues. I hope this will lead to an open discussion on possible strategies and directions to address this problem.

Adina Williams: For Matters Word Order Little MLM.

One possible explanation for the impressive performance of masked language models (MLMs) is that they can learn to represent the syntactic structures prevalent in classical NLP pipelines. Were this correct, we would expect that fine-tuning such models on tasks requiring syntactic structure would lead them to be sensitive to word order at inference time. To address this question, we permute example word order at several steps in the pipeline—during fine-tuning, evaluation, and/or pre-training—and measure the results. We find that permuting word order during fine-tuning has remarkably little effect on downstream performance for several purportedly syntax sensitive NLU tasks (including NLI). Next, we pre-train MLMs on examples with randomly shuffled word order, and find that these models still achieve high accuracy (even after unpermuted fine-tuning) on many downstream tasks—including tasks specifically designed to be challenging for models that ignore word order. Our results show that the success of MLM pre-training is largely due to distributional information not any knowledge of word order per se, and underscores the importance of curating challenging evaluation datasets that require deeper syntactic knowledge.

Table of Contents

Long Papers

WikiBERT Models: Deep Transfer Learning for Many Languages	1
<i>Sampo Pyysalo, Jenna Kanerva, Antti Virtanen and Filip Ginter</i>	
EstBERT: A Pretrained Language-Specific BERT for Estonian	11
<i>Hasan Tanvir, Claudia Kittask, Sandra Eiche and Kairit Sirts</i>	
Operationalizing a National Digital Library: The Case for a Norwegian Transformer Model	20
<i>Per E Kummervold, Javier De la Rosa, Freddy Wetjen and Svein Arne Brygffeld</i>	
Large-Scale Contextualised Language Modelling for Norwegian	30
<i>Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid and Stephan Oepen</i>	
Extremely Low-Resource Machine Translation for Closely Related Languages	41
<i>Maali Tars, Andre Tättar and Mark Fišel</i>	
Measuring Translationese across Levels of Expertise: Are Professionals more Surprising than Students?	53
<i>Yuri Bizzoni and Ekaterina Lapshinova-Koltunski</i>	
CombAlign: a Tool for Obtaining High-Quality Word Alignments	64
<i>Steinþór Steingrímsson, Hrafn Loftsson and Andy Way</i>	
Understanding Cross-Lingual Syntactic Transfer in Multilingual Recurrent Neural Networks	74
<i>Prajit Dhar and Arianna Bisazza</i>	
Speaker Verification Experiments for Adults and Children Using Shared Embedding Spaces	86
<i>Tuomas Kaseva, Hemant Kumar Kathania, Aku Rouhe and Mikko Kurimo</i>	
Spectral Modification for Recognition of Children’s Speech Under Mismatched Conditions	94
<i>Hemant Kumar Kathania, Sudarsana Reddy Kadiri, Paavo Alku and Mikko Kurimo</i>	
A Baseline Document Planning Method for Automated Journalism	101
<i>Leo Leppänen and Hannu Toivonen</i>	
Assessing the Quality of Human-Generated Summaries with Weakly Supervised Learning	112
<i>Joakim Olsen, Arild Brandrud Næss and Pierre Lison</i>	
Knowledge Distillation for Swedish NER models: A Search for Performance and Efficiency	124
<i>Lovisa Hagström and Richard Johansson</i>	
Fine-grained Named Entity Annotation for Finnish	135
<i>Jouni Luoma, Li-Hsin Chang, Filip Ginter and Sampo Pyysalo</i>	
Survey and Reproduction of Computational Approaches to Dating of Historical Texts	145
<i>Sidsel Boldsen and Fredrik Wahlberg</i>	
Multilingual and Zero-Shot is Closing in on Monolingual Web Register Classification	157
<i>Samuel Rönqvist, Valteri Skantsi, Miika Oinonen and Veronika Laippala</i>	
Neural Morphology Dataset and Models for Multiple Languages, from the Large to the Endangered	166

Mika Härmäläinen, Niko Partanen, Jack Rueter and Khalid Alnajjar

CoDeRoMor: A New Dataset for Non-Inflectional Morphology Studies of Swedish	178
<i>Elena Volodina, Yousuf Ali Mohammed and Therese Lindström Tiedemann</i>	
Chunking Historical German	190
<i>Katrin Ortmann</i>	
Part-of-speech Tagging of Swedish Texts in the Neural Era	200
<i>Yvonne Adesam and Aleksandrs Berdicevskis</i>	
De-identification of Privacy-related Entities in Job Postings	210
<i>Kristian Nørgaard Jensen, Mike Zhang and Barbara Plank</i>	
Creating and Evaluating a Synthetic Norwegian Clinical Corpus for De-Identification	222
<i>Synnøve Bråthen, Wilhelm Wie and Hercules Dalianis</i>	
Applying and Sharing Pre-Trained BERT-Models for Named Entity Recognition and Classification in Swedish Electronic Patient Records	231
<i>Mila Grancharova and Hercules Dalianis</i>	
An Unsupervised method for OCR Post-Correction and Spelling Normalisation for Finnish	240
<i>Quan Duong, Mika Härmäläinen and Simon Hengchen</i>	
Learning to Lemmatize in the Word Representation Space	249
<i>Jarkko Lagus and Arto Klami</i>	
Synonym Replacement Based on a Study of Basic-level Nouns in Swedish Texts of Different Complexity	259
<i>Evelina Rennes and Arne Jönsson</i>	
SuperSim: a Test Set for Word Similarity and Relatedness in Swedish	268
<i>Simon Hengchen and Nina Tahmasebi</i>	
NLI Data Sanity Check: Assessing the Effect of Data Corruption on Model Performance	276
<i>Aarne Talman, Marianna Apidianaki, Stergios Chatzikiyiakidis and Jörg Tiedemann</i>	
Finnish Paraphrase Corpus	288
<i>Jenna Kanerva, Filip Ginter, Li-Hsin Chang, Iiro Rastas, Valteri Skantsi, Jemina Kilpeläinen, Hanna-Mari Kupari, Jenna Saarni, Maija Sevón and Otto Tarkka</i>	
Negation in Norwegian: an Annotated Dataset	299
<i>Petter Mæhlum, Jeremy Barnes, Robin Kurtz, Lilja Øvrelid and Erik Velldal</i>	
Short Papers	
What Taggers Fail to Learn, Parsers Need the Most	309
<i>Mark Anderson and Carlos Gómez-Rodríguez</i>	
Investigation of Transfer Languages for Parsing Latin: Italic Branch vs. Hellenic Branch	315
<i>Antonia Karamolegkou and Sara Stymne</i>	
Towards Cross-Lingual Application of Language-Specific PoS Tagging Schemes	321
<i>Hinrik Hafsteinsson and Anton Karl Ingason</i>	
Exploring the Importance of Source Text in Automatic Post-Editing for Context-Aware Machine Translation	326

Chaojun Wang, Christian Hardmeier and Rico Sennrich

Chinese Character Decomposition for Neural MT with Multi-Word Expressions	336
<i>Lifeng Han, Gareth Jones, Alan Smeaton and Paolo Bolzoni</i>	
Grapheme-Based Cross-Language Forced Alignment: Results with Uralic Languages	345
<i>Juho Leinonen, Sami Virpioja and Mikko Kurimo</i>	
Boosting Neural Machine Translation from Finnish to Northern Sámi with Rule-Based Backtranslation	351
<i>Mikko Aulamo, Sami Virpioja, Yves Scherrer and Jörg Tiedemann</i>	
Building a Swedish Open-Domain Conversational Language Model	357
<i>Tobias Norlund and Agnes Stenbom</i>	
It's Basically the Same Language Anyway: the Case for a Nordic Language Model	367
<i>Magnus Sahlgren, Fredrik Carlsson, Fredrik Olsson and Love Börjeson</i>	
Decentralized Word2Vec Using Gossip Learning	373
<i>Abdul Aziz Alkathiri, Lodovico Giaretta, Sarunas Girdzijauskas and Magnus Sahlgren</i>	
Multilingual ELMo and the Effects of Corpus Sampling	378
<i>Vinit Ravishankar, Andrey Kutuzov, Lilja Øvrelid and Erik Velldal</i>	
Should we Stop Training More Monolingual Models, and Simply Use Machine Translation Instead?	385
<i>Tim Isbister, Fredrik Carlsson and Magnus Sahlgren</i>	
Error Analysis of Using BART for Multi-Document Summarization: A Study for English and German Language	391
<i>Timo Johner, Abhik Jana and Chris Biemann</i>	
Grammatical Error Generation Based on Translated Fragments	398
<i>Eetu Sjöblom, Mathias Creutz and Teemu Vahtola</i>	
Creating Data in Icelandic for Text Normalization	404
<i>Helga Svala Sigurðardóttir, Anna Björk Nikulásdóttir and Jón Guðnason</i>	
The Danish Gigaword Corpus	413
<i>Leon Strømberg-Derczynski, Manuel Ciosici, Rebekah Baglini, Morten H. Christiansen, Jacob Aarup Dalsgaard, Riccardo Fusaroli, Peter Juel Henriksen, Rasmus Hvingelby, Andreas Kirkedal, Alex Speed Kjeldsen, Claus Ladefoged, Finn Årup Nielsen, Jens Madsen, Malte Lau Petersen, Jonathan Hvithamar Rystrøm and Daniel Varab</i>	
DanFEVER: Claim Verification Dataset for Danish	422
<i>Jeppe Nørregaard and Leon Derczynski</i>	
The Icelandic Word Web: A Language Technology-Focused Redesign of a Lexicosemantic Database	429
<i>Hjalte Daníelsson, Jón Hilmar Jónsson, Þórður Arnar Árnason, Alec Shaw, Einar Freyr Sigurðsson and Steinþór Steingrímsson</i>	
Getting Hold of Villains and Other Rogues	435
<i>Manfred Klenner, Anne Göhring and Sophia Conrad</i>	
Talrómur: A large Icelandic TTS corpus	440
<i>Atli Sigurgeirsson, Þorsteinn Gunnarsson, Gunnar Örnólfsson, Eydís Magnúsdóttir, Ragnheiður Þórhallsdóttir, Stefán Jónsson and Jón Guðnason</i>	

NorDial: A Preliminary Corpus of Written Norwegian Dialect Use	445
<i>Jeremy Barnes, Petter Mæhlum and Samia Touileb</i>	
The Swedish Winogender Dataset	452
<i>Saga Hansson, Konstantinos Mavromatakis, Yvonne Adesam, Gerlof Bouma and Dana Dannélls</i>	
 Demo Papers	
DaNLP: An Open-Source Toolkit for Danish Natural Language Processing	460
<i>Amalie Brogaard Pauli, Maria Barrett, Ophélie Lacroix and Rasmus Hvingelby</i>	
HB Deid - HB De-Identification Tool Demonstrator	467
<i>Hanna Berg and Hercules Dalianis</i>	

Long Papers

WikiBERT Models: Deep Transfer Learning for Many Languages

Sampo Pyysalo, Jenna Kanerva, Antti Virtanen, Filip Ginter

TurkuNLP group,
Department of Computing,
Faculty of Technology
University of Turku, Finland
first.last@utu.fi

Abstract

Deep neural language models such as BERT have enabled substantial advances in natural language processing. However, due to the effort and computational cost involved in their pre-training, such models are typically introduced only for high-resource languages. In this paper, we introduce a simple, fully automated pipeline for creating language-specific BERT models from Wikipedia data and introduce 42 new monolingual models, most for languages up to now lacking such resources. We show that the newly introduced WikiBERT models outperform multilingual BERT (mBERT) in cloze tests for nearly all languages, and that parsing using WikiBERT models outperforms mBERT on average, with substantially improved performance for some languages, but decreases for others. All of the resources introduced in this work are available under open licenses from <https://github.com/turkunlp/wikibert>.

1 Introduction

Transfer learning using language models pre-trained on large unannotated corpora has allowed for substantial recent advances at a broad range of natural language processing (NLP) tasks. By contrast to earlier distributional semantics approaches such as random indexing (Kanerva et al., 2000) and context-independent neural approaches such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), models such as ULMFiT (Howard and Ruder, 2018), ELMo (Peters et al., 2018), GPT (Radford et al., 2018) and BERT (Devlin et al., 2019) create contextualized representations of meaning, capable of providing both contextualized word embeddings as well as embed-

dings for text segments longer than words. Recent pre-trained neural language models have been rapidly advancing the state of the art in a range of natural language understanding and NLP tasks (Wang et al., 2018, 2019; Straková et al., 2019; Kondratyuk and Straka, 2019).

The transformer architecture (Vaswani et al., 2017) and the BERT language model of Devlin et al. (2019) have been particularly influential, with transformer-based models in general and BERT in particular fuelling a broad range of advances and serving as the basis of many recent studies of neural language models (e.g. Lan et al., 2019; Liu et al., 2019; Sanh et al., 2019). As is the case for most studies on new deep neural language models, the original study introducing BERT addressed only English. The authors later released a Chinese model as well as a multilingual model, mBERT, trained on text from 104 languages, but opted not to introduce models specifically targeting other languages. While mBERT is a powerful multilingual model with remarkable cross-lingual capabilities (Pires et al., 2019), it remains a compromise in that the 104 languages share the model capacity dedicated to one language in monolingual models, and it consequently suffers from degradation of performance in language-specific tasks (Conneau et al., 2020).

Here, we take steps towards closing various parts of the gap between languages with dedicated deep neural models, ones that share capacity with others in a massively multilingual model, and ones that lack any representation at all. We introduce a fully automated pipeline for creating language-specific BERT models from Wikipedia data and apply this pipeline to create 42 new such models.

2 Related work

Considerable recent effort by various groups has focused on introducing dedicated BERT models covering single languages or a small num-

ber of (often closely related) languages. Dedicated monolingual models include e.g. BERTje¹ (de Vries et al., 2019) for Dutch, CamemBERT² (Martin et al., 2020) for French, FinBERT³ (Virtanen et al., 2019) for Finnish, RuBERT⁴ (Kurato and Arkhipov, 2019) for Russian, and Romanian BERT (Dumitrescu et al., 2020); more focused multilingual models include e.g. the bilingual Finnish-English model of Chang et al. (2020) and the trilingual Finnish-Estonian-English and Croatian-Slovenian-English models of Ulčar and Robnik-Šikonja (2020).

Many of these studies have demonstrated the newly introduced models to allow for substantial improvements over mBERT in various language-specific downstream task evaluations, thus supporting the continued value of creating monolingual and focused multilingual models. However, these efforts still cover only a fairly limited number of languages, and do not offer a straightforward way to substantially extend that coverage. The studies further differ considerably in aspects such as data collection, text cleaning and preprocessing, pre-training parameter setting and other details of the pre-training process, making it difficult to meaningfully compare the models to address questions such as *which languages benefit most from mono/multilingual pre-training?* We are not aware of previous efforts to automate the creation of large numbers of monolingual deep neural language models from comparable, publicly available sources nor efforts to create broad-coverage collections of such models.

In a line of study in some senses orthogonal to our work, a number of massively multilingual models improving on mBERT in terms of model architecture, training dataset, objectives, and process or other aspects have been introduced (e.g. Conneau et al., 2020; Xue et al., 2020). While it is certainly an interesting question to ask what the tradeoffs between monolingual and massively multilingual pre-training are for models other than BERT, it is not feasible for us to replicate the training processes for other models, and we have here chosen to focus on BERT-based models and Wikipedia due to their prominence and status as benchmarks.

¹<https://github.com/wietsedv/bertje>

²<https://camembert-model.fr/>

³<https://turkunlp.org/FinBERT/>

⁴<https://github.com/deepmipt/deeppavlov/>

3 Data

We next introduce the two primary datasets used in this study: Wikipedia, used as the source of unannotated texts for model pre-training, and Universal Dependencies annotated corpora, used to train preprocessing methods as well as in model evaluation.

3.1 Wikipedia

Wikipedia is a collaboratively created online encyclopedia that is available in a large number of languages under open data licenses. The English Wikipedia was the main source of text for pre-training the original English BERT models, accounting for three-fourths of its pre-training data.⁵ The mBERT models were likewise trained exclusively on Wikipedia data. In this work, we chose to focus on the Wikipedias in various languages as the only source of pre-training data, thus assuring that our approach can be directly applied to a broad selection of languages and providing direct comparability with existing models, in particular mBERT.

As of this writing, the List of Wikipedias⁶ identifies Wikipedias in 309 languages. Their sizes vary widely: while the largest of the set, the English Wikipedia, contains over six million articles, the smaller half of Wikipedias (155 languages) put together only total approximately 400,000 articles. As the BERT base model has over 100 million parameters and BERT models are frequently trained on billions of words of unannotated text, it seems safe to estimate that attempting to train BERT with the data from one of the smaller wikipedias⁷ would likely not produce a very successful model. It is nevertheless not well established how much unannotated text is required to pre-train a language-specific model, and how much the domain and quality of the pre-training data affect the model performance.

In order to focus computational resources on models with practical value, we opted to exclude “dead” languages that are not in everyday spoken use by any community from our efforts. We have

⁵The remaining quarter of BERT pre-training data was drawn from the BooksCorpus (Zhu et al., 2015), a unique (and now unavailable) resource for which analogous resources in other languages cannot be readily created.

⁶https://en.wikipedia.org/wiki/List_of_Wikipedias

⁷For example, Old Church Slavonic, ranked 272nd among wikipedias by size, has fewer than 1000 articles and under 50,000 tokens.

Language (code)	Tokens	Language (code)	Tokens	Language (code)	Tokens
Afrikaans (af)	24M	Finnish (fi)	97M	Norwegian (no)	112M
Arabic (ar)	184M	French (fr)	858M	Polish (pl)	282M
Belarusian (be)	34M	Galician (gl)	58M	Portuguese (pt)	326M
Bulgarian (bg)	71M	Hebrew (he)	166M	Romanian (ro)	85M
Catalan (ca)	236M	Hindi (hi)	35M	Russian (ru)	565M
Czech (cs)	143M	Croatian (hr)	54M	Slovak (sk)	39M
Danish (da)	65M	Hungarian (hu)	129M	Slovenian (sl)	42M
German (de)	1.0B	Indonesian (id)	93M	Serbian (sr)	96M
Greek (el)	81M	Italian (it)	579M	Swedish (sv)	364M
English (en)	2.7B	Japanese (ja)	596M	Tamil (ta)	26M
Spanish (es)	678M	Korean (ko)	79M	Turkish (tr)	71M
Estonian (et)	38M	Lithuanian (lt)	34M	Ukrainian (uk)	260M
Basque (eu)	45M	Latvian (lv)	21M	Urdu (ur)	18M
Persian (fa)	95M	Dutch (nl)	300M	Vietnamese (vi)	172M

Table 1: Wikipedia sizes for selected languages.

otherwise broadly proceeded to introduce preprocessing support and models for languages in decreasing order of the size of their Wikipedias and support in Universal Dependencies, discussed below. Table 1 lists the Wikipedias used in this work.

3.2 Universal Dependencies

Universal Dependencies (UD) is a community-lead effort aiming to create cross-linguistically consistent treebank annotations for many typologically different languages (Nivre et al., 2016, 2020). In this study, we rely on UD both as training data for components of the preprocessing pipeline (Section 4.1) as well as for our evaluations. As of this writing, the latest release of the UD treebanks⁸ is 2.7, which includes 183 treebanks covering 104 languages, thus matching mBERT in terms of the raw number of covered languages.

To maintain comparability with recent work on UD parsing, we use the UD v2.3 treebanks,⁹ with 129 treebanks in 76 languages, in our comparative experiments assessing the WikiBERT models. We further limit our evaluation to the subset of UD v2.3 treebanks that have training, development, and test sets, thus excluding e.g. the 17 parallel UD treebanks which only provide test sets. We further exclude from evaluation treebanks released without text (ar_nyuad, en_esl, fr_ftb, ja_bccwj), the Swedish sign language treebank (swl_sslc), and treebanks in languages for which

we have not trained dedicated models (mr_ufal, mt_mudt, te_mtg, and ug_udt). Table 2 lists the treebanks applied in our evaluation. We note that there is very substantial variance between treebanks in the amount of training data available, ranging from little over 3000 tokens for the Lithuanian HSE treebank to more than a million for the Czech PDT.

4 Methods

We next briefly introduce the primary steps of the preprocessing pipeline for creating pre-training examples from Wikipedia source as well as the tools used for text processing, model pre-training, and evaluation. We refer to our published pipeline and its documentation for full processing details.

4.1 Preprocessing pipeline

In order to create high quality pre-training data from raw Wikipedia dumps in the format required by BERT model training, we introduce a pipeline that performs the following primary steps:

Data and model download The full Wikipedia database backup dump is downloaded from a mirror site¹⁰ and a UDPipe model for the language from the LINDAT/CLARIN repository.¹¹

Plain text extraction WikiExtractor¹² is used to extract plain text with document boundaries from the Wikipedia XML dump.

⁸<https://universaldependencies.org/>

⁹<http://hdl.handle.net/11234/1-2895>

¹⁰<https://dumps.wikimedia.org/>

¹¹<http://hdl.handle.net/11234/1-3131>

¹²<https://github.com/attardi/wikiextractor>

Language (code)	Treebank	Tokens	Language (code)	Treebank	Tokens
Afrikaans (af)	AfriBooms	33894	Indonesian (id)	GSD	97531
Arabic (ar)	PADT	223881	Italian (it)	ISDT	276019
Belarusian (be)	HSE	5217	Italian (it)	ParTUT	48934
Bulgarian (bg)	BTB	124336	Italian (it)	PoSTWITA	99441
Catalan (ca)	AnCora	417587	Japanese (ja)	GSD	160419
Czech (cs)	CAC	472609	Korean (ko)	GSD	56687
Czech (cs)	CLTT	26742	Korean (ko)	Kaist	296446
Czech (cs)	FicTree	133637	Lithuanian (lt)	HSE	3210
Czech (cs)	PDT	1173282	Latvian (lv)	LVTB	113405
Danish (da)	DDT	80378	Dutch (nl)	Alpino	186046
German (de)	GSD	263804	Dutch (nl)	LassySmall	75134
Greek (el)	GDT	42326	Norwegian (no)	Bokmaal	243887
English (en)	EWT	204585	Norwegian (no)	Nynorsk	245330
English (en)	GUM	53686	Polish (pl)	LFG	104750
English (en)	LinES	50091	Polish (pl).	SZ	62501
English (en)	ParTUT	43518	Portuguese (pt)	Bosque	206744
Spanish (es)	AnCora	444617	Portuguese (pt)	GSD	255755
Spanish (es)	GSD	382436	Romanian (ro)	Nonstandard	155498
Estonian (et)	EDT	341122	Romanian (ro)	RRT	185113
Basque (eu)	BDT	72974	Russian (ru)	GSD	75964
Persian (fa)	Seraji	121064	Russian (ru)	SynTagRus	870474
Finnish (fi)	FTB	127602	Slovak (sk)	SNK	80575
Finnish (fi)	TDT	162621	Slovenian (sl)	SSJ	112530
French (fr)	GSD	354699	Serbian (sr)	SET	65764
French (fr)	ParTUT	24123	Swedish (sv)	LinES	48320
French (fr)	Sequoia	50536	Swedish (sv)	Talbanken	66645
French (fr)	Spoken	14952	Tamil (ta)	TTB	6329
Galician (gl)	CTG	79327	Turkish (tr)	IMST	37918
Hebrew (he)	HTB	137721	Ukrainian (uk)	IU	88043
Hindi (hi)	HDTB	281057	Urdu (ur)	UDTB	108690
Croatian (hr)	SET	154055	Vietnamese (vi)	VTB	20285
Hungarian (hu)	Szeged	20166			

Table 2: UD v2.3 training data sizes for selected treebanks.

Segmentation and tokenization UDPipe is used with the downloaded model to segment sentences and tokenize the plain text, producing text with document, sentence, and word boundaries.

Document filtering A set of heuristic rules and statistical language detection¹³ are applied to optionally filter documents based on configurable criteria.¹⁴

Sampling and basic tokenization A sample of sentences is tokenized using BERT basic tokeniza-

tion¹⁵ to produce examples for vocabulary generation that match BERT tokenization criteria.

Vocabulary generation A subword vocabulary is generated using the SentencePiece¹⁶ (Kudo and Richardson, 2018) implementation of byte-pair encoding (Gage, 1994; Sennrich et al., 2015). After generation the vocabulary is converted to the BERT WordPiece format (a different but largely equivalent representation).

¹³<https://github.com/shuyo/language-detection>

¹⁴We note that there are Wikipedia pages whose content is mostly in a language different from that of the Wikipedia.

¹⁵BERT basic tokenization preserves alphanumeric sequences but separates e.g. all punctuation characters into individual tokens.

¹⁶<https://github.com/google/sentencepiece>

Language (code)	Subword Accuracy		Language (code)	Subword Accuracy	
	mBERT	WikiBERT		mBERT	WikiBERT
Afrikaans (af)	28.69	43.22	Indonesian (id)	30.72	52.47
Arabic (ar)	20.17	29.96	Italian (it)	29.48	37.98
Belarusian (be)	18.15	36.39	Japanese (ja)	49.25	45.19
Bulgarian (bg)	21.26	39.98	Korean (ko)	17.59	30.61
Catalan (ca)	40.29	56.63	Lithuanian (lt)	15.11	29.83
Czech (cs)	22.41	39.77	Latvian (lv)	15.59	29.99
Danish (da)	25.06	40.86	Dutch (nl)	29.08	47.54
German (de)	33.85	46.93	Norwegian (no)	22.73	34.15
Greek (el)	21.42	45.42	Polish (pl)	17.64	33.30
English (en)	37.39	46.64	Portuguese (pt)	32.55	43.85
Spanish (es)	40.20	52.05	Romanian (ro)	21.19	33.07
Estonian (et)	14.00	31.26	Russian (ru)	27.16	46.86
Basque (eu)	15.15	30.99	Slovak (sk)	16.52	29.08
Persian (fa)	21.52	45.20	Slovenian (sl)	21.21	35.24
Finnish (fi)	12.89	27.67	Serbian (sr)	25.80	30.70
French (fr)	41.30	52.08	Swedish (sv)	22.11	37.11
Galician (gl)	33.23	36.81	Tamil (ta)	14.36	31.85
Hebrew (he)	20.96	21.83	Turkish (tr)	12.56	29.16
Hindi (hi)	19.97	47.23	Ukrainian (uk)	19.15	31.78
Croatian (hr)	23.03	39.99	Urdu (ur)	20.83	39.70
Hungarian (hu)	18.89	38.99	Vietnamese (vi)	17.96	47.35

Table 3: Results for the cloze test in terms of subword prediction accuracy (percentages)

Example generation Masked language modeling and next sentence prediction examples using the full BERT tokenization specified by the generated vocabulary are created in the TensorFlow TFRecord format using BERT tools.

The created vocabulary and pre-training examples can be used directly with the original BERT implementation to train new language-specific models.

4.2 UDPipe

UDPipe (Straka et al., 2016) is a parser capable of producing segmentation, part-of-speech and morphological tags, lemmas and dependency trees. In this work we use UDPipe for sentence segmentation and tokenization in the preprocessing pipeline. The segmentation component in UDPipe is a character-level bidirectional GRU network simultaneously predicting the end-of-token and end-of-sentence markers.

4.3 Pre-training

We aimed to largely mirror the original BERT process in our selection of parameters and settings for the pre-training process to create the WikiBERT models, with some adjustments made to ac-

count for differences in computational resources. Specifically, while the original BERT models were trained on TPUs, we trained on Nvidia Volta V100 GPUs with 32GB memory. We followed the original BERT processing in training for a total of 1M steps in two stages, the first 900K steps with a maximum sequence length of 128, and the last 100K steps with a maximum of 512. Due to memory limitations, each model was trained on 4 GPUs using a batch size of 140 during the sequence length 128 phase, and 8 GPUs with a batch size of 20 during the sequence length 512 phase.

4.4 Cloze test

In order to evaluate the BERT models with respect to their original training objective, we employ a cloze test, where words are randomly masked and predicted back. We mask a random 15% of words in each sentence, and, in case a word is composed of several subword (WordPiece) tokens, all subword tokens are masked for an easier and more meaningful evaluation (cf. full-word masking in BERT pre-training). All masked positions are predicted at once in the same manner as done in the BERT pre-training (i.e. without iterative predic-

Language (code)	Average LAS		Language (code)	Average LAS	
	mBERT	WikiBERT		mBERT	WikiBERT
Afrikaans (af)	87.85	87.33	Indonesian (id)	80.40	80.12
Arabic (ar)	83.81	85.47	Italian (it)	89.64	89.77
Belarusian (be)	81.77	79.81	Japanese (ja)	92.78	92.92
Bulgarian (bg)	92.30	92.51	Korean (ko)	86.19	87.28
Catalan (ca)	92.08	92.06	Lithuanian (lt)	58.68	58.40
Czech (cs)	90.45	90.69	Latvian (lv)	84.29	84.46
Danish (da)	85.78	85.84	Dutch (nl)	90.26	91.02
German (de)	83.16	84.13	Norwegian (no)	91.54	91.94
Greek (el)	91.63	92.35	Polish (pl)	94.45	95.58
English (en)	88.09	88.05	Portuguese (pt)	91.91	92.21
Spanish (es)	90.42	90.12	Romanian (ro)	86.83	86.52
Estonian (et)	85.86	87.43	Russian (ru)	90.35	91.13
Basque (eu)	82.99	83.70	Slovak (sk)	91.64	91.73
Persian (fa)	86.60	88.60	Slovenian (sl)	92.83	93.37
Finnish (fi)	87.64	90.81	Serbian (sr)	92.30	91.79
French (fr)	89.22	88.77	Swedish (sv)	86.42	87.12
Galician (gl)	83.05	82.61	Tamil (ta)	70.14	69.63
Hebrew (he)	88.77	90.17	Turkish (tr)	69.33	71.25
Hindi (hi)	91.59	91.86	Ukrainian (uk)	88.57	90.41
Croatian (hr)	89.46	89.40	Urdu (ur)	82.66	82.15
Hungarian (hu)	83.99	86.21	Vietnamese (vi)	66.89	68.87

Table 4: Average LAS results for UDify for Universal Dependencies treebanks in each language.

tion of one position per time step). As a source of sentences, we use the first 1000 sentences of training sections of the treebanks, limited to sentences of 5–50 tokens in length. We note that the treebanks are not entirely non-overlapping with Wikipedia: 16 out of the 63 treebanks draw at least part of their texts from Wikipedia. However, as all of the compared models share this source of pre-training data, we do not expect this overlap to bias the comparison.

4.5 UDify

To assess the performance of the models in a downstream task, we apply the UDify parser (Kondratyuk and Straka, 2019), initialized with one of the models and trained on Universal Dependencies data. UDify is a state-of-the-art model and can predict UD part-of-speech tags, morphological features, lemmas, and dependency trees. UDify implements a multi-task learning objective using task-specific prediction layers on top of a pre-trained BERT encoder. All prediction layers are trained simultaneously, while also fine-tuning the pre-trained encoder weights. In the following evaluation, we focus on the parsing per-

formance using the standard Labeled Attachment Score (LAS) metric.

5 Results

We next present the results of the intrinsic cloze test evaluation and the extrinsic evaluation with syntactic analysis as a downstream task.

5.1 Cloze evaluation results

The cloze evaluation results are shown in Table 3, where we measure subword-level prediction accuracy, i.e. the proportion of cases where the model assigns the highest probability to the original subword. We find that the WikiBERT models outperform mBERT for all languages except for Japanese,¹⁷ averaging more than 10% points higher accuracy. While this is an encouraging result regarding the quality of the newly introduced models, the evaluation is arguably biased in favour of monolingual models, as their candidate space (the vocabulary) is limited to only include options in the correct language. More broadly, success at

¹⁷This result may suggest some issues specific to Japanese either in the preprocessing pipeline or the applied UDify model, but we have yet to identify any clear explanation for the exception.

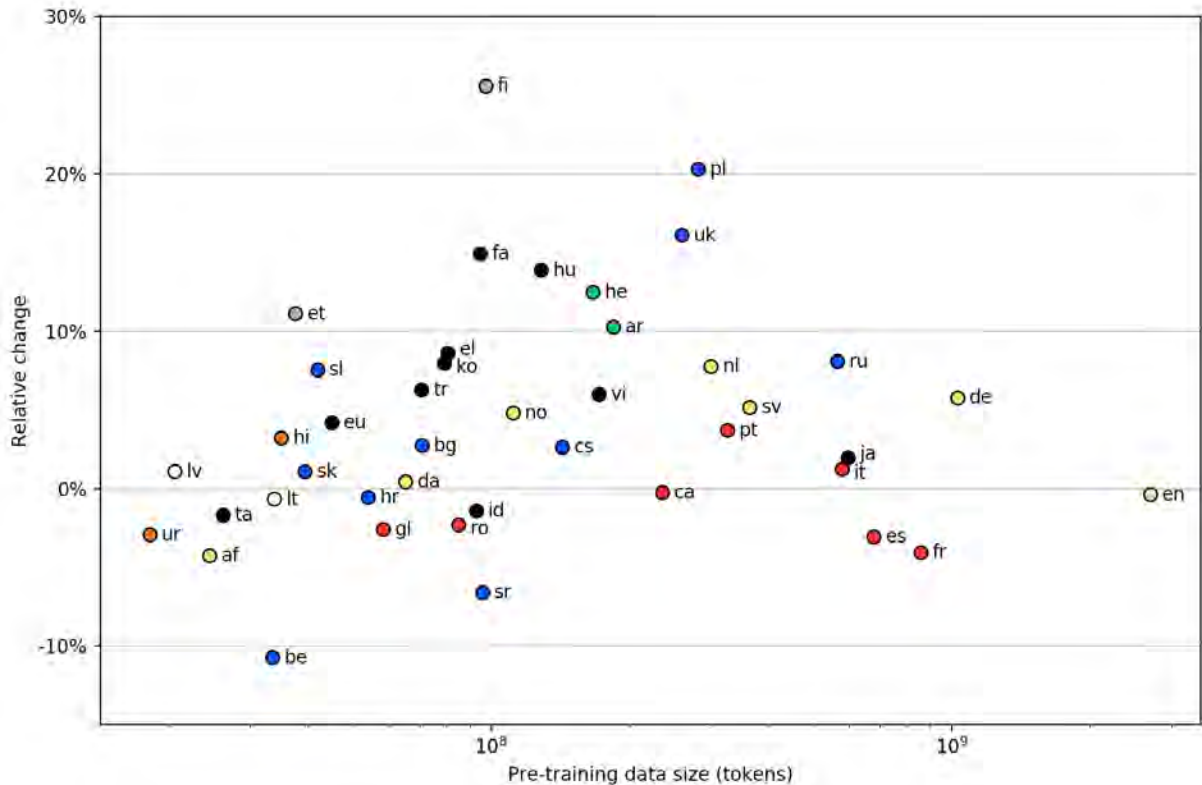


Figure 1: Average relative change in LAS when replacing mBERT with a WikiBERT model for UDify initialization plotted against the WikiBERT pre-training data size in tokens. Coloring indicates language grouping by genera (Baltic: white, Finnic: light blue, Germanic: yellow, Indic: orange, Romance: red, Semitic: green, Slavic: blue, other: black).

intrinsic evaluations such as this does not guarantee practical applicability (or vice versa), and models should also be assessed at real-world tasks to gain a more complete picture of their value (see e.g. Chiu et al., 2016).

5.2 UD parsing results

Table 4 summarizes the results of the UD parsing evaluation. Given the large size of both train sets (See Table 2) and test sets for most of the languages, the evaluation results are stable, and we have found that repetitions of the training process often result in less than 0.1% point differences between runs. To conserve computational resources, we have thus here chosen to run a single experiment per treebank (a typical setting for UD evaluation).

We find a complex, mixed picture where mBERT and WikiBERT models each appear clearly superior for different languages, for example, mBERT for Belarusian and WikiBERT for Finnish. On average across all languages, UDify with WikiBERT models slightly edges out UDify

with mBERT, with an 86.1% average for mBERT and 86.6% for WikiBERT (an approximately 4% relative decrease in LAS error). However, such averaging hides more than it reveals, and it is more interesting to consider the various potential impacts on performance from pre-training data size, potential support from close relatives in the same language family, and other similar factors. The various UD treebanks represent very different levels of challenge with LAS results ranging from below 60% to above 95%, and to reduce the impact of the properties of the treebanks on the comparison, in the following we focus on the relative change in performance when initializing UDify with a WikiBERT model compared to the baseline approach using mBERT.

Figure 1 shows the average relative change in performance over all treebanks for a language when replacing mBERT with the relevant WikiBERT model for UDify, plotted against the number of tokens in Wikipedia for the language. While the data is very noisy due to a number of factors, we find some indication of a “sweet spot”

where training a dedicated monolingual model tends to show most benefit over using the multilingual model when at least approximately 100M tokens but fewer than 1B tokens of pre-training data are available. We also briefly note some other properties in this data:

- For English, a language in the large Germanic family and the one with the largest amount of Wikipedia pre-training data, mBERT and WikiBERT results are effectively identical.
- The greatest loss when moving from mBERT to a WikiBERT model is seen for Belarusian, a slavic language closely related to Russian, for which considerably more pre-training data is available.
- The greatest gain when moving from mBERT to a WikiBERT model is seen for Finnish, a Finnic language with few closely related, widely spoken languages, which has a comparatively large Wikipedia.

Observations such as these may suggest fruitful avenues for further research into the conditions under which mono- and multilingual language model training is expected to be most successful. Based on these results and the findings of studies training models for small numbers of closely related languages (see Section 2), we anticipate that multilingual training may most readily benefit lower-resourced languages trained together with a closely related high-resource language in a bilingual setting.

6 Discussion and conclusions

In this paper, we have introduced a simple, fully automatic pipeline for creating monolingual BERT models from Wikipedia data, applied the pipeline to introduce 42 new language-specific models, most covering languages that previously lacked a dedicated deep neural language model. We evaluated the WikiBERT models intrinsically using cloze evaluation, finding that they outperform the multilingual mBERT model for all but one language. An extrinsic evaluation using a dependency parsing task with Universal Dependencies data and the UDify neural parser found a more nuanced picture of the comparative merits of the monolingual and multilingual models: while we found that a WikiBERT model will provide better performance than mBERT on average and in

multiple cases provides a more than 10% relative decrease in LAS error compared to the multilingual model, the WikiBERT models showed lower performance than mBERT for multiple languages. Viewing relative change in performance against pre-training data size, we found indications that monolingual models may most benefit languages that have no closely related high-resource languages and for which comparatively large pre-training corpora can be assembled.

The availability of the WikiBERT collection of models opens up a broad range of potential avenues for research into the strengths, weaknesses and challenges in both mono- and multilingual language modeling that we hope to pursue in future work. We also hope to encourage both monolingual applications as well as exploration of these questions by others by making the models freely available under open licenses from <https://github.com/turkunlp/wikibert>.

Acknowledgments

This work was funded in part by the Academy of Finland. We wish to thank CSC – IT Center for Science, Finland, for providing generous computational resources for this study.

References

- Li-Hsin Chang, Sampo Pyysalo, Jenna Kanerva, and Filip Ginter. 2020. Towards fully bilingual deep language modeling. *arXiv preprint arXiv:2010.11639*.
- Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 1–6.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

- Stefan Dumitrescu, Andrei-Marius Avram, and Sampo Pyysalo. 2020. The birth of Romanian BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4324–4328.
- Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.
- Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.
- Pentti Kanerva, Jan Kristoferson, and Anders Holst. 2000. Random indexing of text samples for latent semantic analysis. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 22.
- Daniel Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. *arXiv preprint arXiv:1904.02099*.
- Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.
- Yuri Kuratov and Mikhail Arhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *arXiv preprint arXiv:1905.07213*.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villamonte de la Clergerie, Djamel Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.
- Tomas Mikolov, G.s Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, pages 1–12.
- Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 1659–1666.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. *arXiv preprint arXiv:2004.10643*.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? *arXiv preprint arXiv:1906.01502*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. *Technical report, OpenAI*.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipeline: trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 4290–4297.
- Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331.
- Matej Ulčar and Marko Robnik-Šikonja. 2020. FinEst BERT and CroSloEngual BERT: less is more in multilingual models. *arXiv preprint arXiv:2006.07890*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for finnish. *arXiv preprint arXiv:1912.07076*.

- Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A dutch BERT model. *arXiv preprint arXiv:1912.09582*.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, pages 3261–3275.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

EstBERT: A Pretrained Language-Specific BERT for Estonian

Hasan Tanvir and Claudia Kittask and Sandra Eiche and Kairit Sirts

Institute of Computer Science

University of Tartu

Tartu, Estonia

hasantanvir79@gmail.com, {claudia.kittask,sandra.eiche,sirts}@ut.ee

Abstract

This paper presents EstBERT, a large pre-trained transformer-based language-specific BERT model for Estonian. Recent work has evaluated multilingual BERT models on Estonian tasks and found them to outperform the baselines. Still, based on existing studies on other languages, a language-specific BERT model is expected to improve over the multilingual ones. We first describe the EstBERT pretraining process and then present the models' results based on the finetuned EstBERT for multiple NLP tasks, including POS and morphological tagging, dependency parsing, named entity recognition and text classification. The evaluation results show that the models based on EstBERT outperform multilingual BERT models on five tasks out of seven, providing further evidence towards a view that training language-specific BERT models are still useful, even when multilingual models are available.¹

1 Introduction

Pretrained language models, such as BERT (Devlin et al., 2019) or ELMo (Peters et al., 2018), have become the essential building block for many NLP systems. These models are trained on large amounts of unannotated textual data, enabling them to capture the general regularities in the language and thus can be used as a basis for training the subsequent models for more specific NLP tasks. Bootstrapping NLP systems with pretraining is particularly relevant and holds the greatest promise for improvements in the setting of limited resources, either when working with tasks of limited annotated training data or less-resourced languages like Estonian.

Since the first publication and release of the large pretrained language models on English, considerable effort has been made to develop support for other languages. In this regard, multilingual BERT models, simultaneously trained on the text of many different languages, have been published, several of which also in-

clude the Estonian language (Devlin et al., 2019; Conneau et al., 2019; Sanh et al., 2019; Conneau and Lample, 2019). These multilingual models' performance was recently evaluated on several Estonian NLP tasks, including POS and morphological tagging, named entity recognition, and text classification (Kittask et al., 2020). The overall conclusions drawn from these experiments are in line with previously reported results on other languages, i.e., for many or even most tasks, multilingual BERT models help improve performance over baselines that do not use language model pretraining.

Besides multilingual models, language-specific BERT models have been trained for an increasing number of languages, including for instance CamBERT (Martin et al., 2020) and FlauBERT (Le et al., 2020) for French, FinBERT for Finnish (Virtanen et al., 2019), RobBERT (Delobelle et al., 2020) and BERTJe (de Vries et al., 2019) for Dutch, Chinese BERT (Cui et al., 2019), BETO for Spanish (Cañete et al., 2020), RuBERT for Russian (Kuratov and Arkhipov, 2019) and others. For a recent overview about these efforts refer to Nozza et al. (2020). Aggregating the results over different language-specific models and comparing them to those obtained with multilingual models shows that depending on the task, the average improvement of the language-specific BERT over the multilingual BERT varies from 0.70 accuracy points in paraphrase identification up to 6.37 in sentiment classification (Nozza et al., 2020). The overall conclusion one can draw from these results is that while existing multilingual BERT models can bring along improvements over language-specific baselines, using language-specific BERT models can further considerably improve the performance of various NLP tasks.

Following the line of reasoning presented above, we set forth to train EstBERT, a language-specific BERT model for Estonian. In the following sections, we first give details about the data used for BERT pretraining and then describe the pretraining process. Finally, we will provide evaluation results on the same tasks as presented by Kittask et al. (2020) on multilingual BERT models, which include POS and morphological tagging, named entity recognition and text classification. Additionally, we also train a dependency parser based on the spaCy system. Compared to multilingual

¹The model is available via HuggingFace Transformers library: <https://huggingface.co/tartuNLP/EstBERT>

models, the EstBERT model achieves better results on five tasks out of seven, providing further evidence for the usefulness of pretraining language-specific BERT models. Additionally, we also compare with the Estonian WikiBERT, a recently published Estonian-specific BERT model trained on a relatively small Wikipedia data (Pyysalo et al., 2020). Compared to the Estonian WikiBERT model, the EstBERT achieves better results on six tasks out of seven, demonstrating the positive effect of the amount of pretraining data on the generalisability of the model.

2 Data Preparation

The first step for training the EstBERT model involves preparing a suitable unlabeled text corpus. This section describes both the steps we took to clean and filter the data and the process of generating the vocabulary and the pretraining examples.

2.1 Data Preprocessing

For training the EstBERT model, we used the Estonian National Corpus 2017 (Kallas and Koppel, 2018),² which was the largest Estonian language corpus available at the time. It consists of four sub-corpora: the Estonian Reference Corpus 1990-2008, the Estonian Web Corpus 2013, the Estonian Web Corpus 2017, and the Estonian Wikipedia Corpus 2017. The Estonian Reference corpus (ca 242M words) consists of a selection of electronic textual material, about 75% of the corpus contains newspaper texts, the rest 25% contains fiction, science and legislation texts. The Estonian Web Corpora 2013 and 2017 make up the largest part of the material and they contain texts collected from the Internet. The Estonian Wikipedia Corpus 2017 is the Estonian Wikipedia dump downloaded in 2017 and contains roughly 38M words. The top row of the Table 1 shows the initial statistics of the corpus.

We applied different cleaning and filtering techniques to preprocess the data. First, we used the corpus processing methods from EstNLTK (Laur et al., 2020), which is an open-source tool for Estonian natural language processing. Using the EstNLTK, all XML/HTML tags were removed from the text, also all documents with a language tag other than Estonian were removed. Additional non-Estonian documents were further filtered out using the language-detection library.³ Next, all duplicate documents were removed. For that, we used hashing—all documents were lowercased, and then the hashed value of each document was subsequently stored into a set. Only those documents whose hash value did not yet exist in the set (i.e., the first document with each hash value) were retained. We also used the hand-written heuristics,⁴ developed for preprocessing the data for training the FinBERT model

²<https://www.sketchengine.eu/estonian-national-corpus/>

³<https://github.com/shuyo/language-detection>

⁴<https://github.com/TurkuNLP/deepfin-tools>

	Documents	Sentences	Words
Initial	3.9M	87.6M	1340M
After cleanup	3.3M	75.7M	1154M

Table 1: Statistics of the corpus before and after the cleanup.

(Virtanen et al., 2019), to filter out documents with too few words, too many stopwords or punctuation marks, for instance. We applied the same thresholds as were used for Finnish BERT. Finally, the corpus was truecased by lemmatizing a copy of the corpus with EstNLTK tools and using the lemma’s casing information to decide whether the word in the original corpus should be upper- or lowercase. The statistics of the corpus after the preprocessing and cleaning steps are in the bottom row of Table 1.

2.2 Vocabulary and Pretraining Example Generation

Originally, BERT uses the WordPiece tokeniser, which is not available open-source. Instead, we used the BPE tokeniser available in the open-source sentencepiece⁵ library, which is the closest to the WordPiece algorithm, to construct a vocabulary of 50K subword tokens. Then, we used BERT tools⁶ to create the pretraining examples for the BERT model in the TFRecord format. In order to enable parallel training on four GPUs, the data was split into four shards. Separate pretraining examples with sequences of length 128 and 512 were created, masking 15% of the input words in both cases. Thus, 20 and 77 words in maximum were masked in sequences of both lengths, respectively.

3 Evaluation Tasks

Before describing the EstBERT model pretraining process itself, we first describe the tasks used to both validate and evaluate our model. These tasks include the POS and morphological tagging, named entity recognition, and text classification. In the following subsection, we describe the available Estonian datasets for these tasks.

3.1 Part of Speech and Morphological Tagging

For part of speech (POS) and morphological tagging, we use the Estonian EDT treebank from the Universal Dependencies (UD) collection that contains annotations of lemmas, parts of speech, universal morphological features, dependency heads, and universal dependency labels. We use the UD version 2.5 to enable comparison with the experimental results of the multilingual BERT models reported by Kittask et al. (2020). We train models to predict both universal POS (UPOS) and language-specific POS (XPOS) tags as well as

⁵<https://github.com/google/sentencepiece>

⁶<https://github.com/google-research/bert>

morphological features. The pre-defined train/dev/test splits are used for training and evaluation. Table 2 shows the statistics of the treebank splits. The accuracy of the POS and morphological tagging tasks is evaluated with the `con1118_ud_eval` script from the CoNLL 2018 Shared Task.

	Train	Dev	Test
Sentences	31012	3128	6348
Tokens	344646	42722	48491

Table 2: Statistics of the UDv2.5 Estonian treebank.

3.2 Named Entity Recognition

Estonian named entity recognition (NER) corpus (Tkachenko et al., 2013) annotations cover three types of named entities: locations, organizations, and persons. It contains 572 news stories published in local online newspapers Postimees and Delfi, covering local and international news on various topics. Table 3 displays statistics of the training, development and test splits. The performance of the NER models is evaluated with the `con11eval` script from the CoNLL 2000 shared task.

	Tokens	PER	LOC	ORG	Total
Train	155981	6174	4749	4784	15707
Dev	32890	1115	918	742	2775
Test	28370	1201	644	619	2464

Table 3: Statistics of the Estonian NER corpus.

3.3 Sentiment and Rubric Classification

Estonian Valence corpus (Pajupuu et al., 2016) consists of 4085 news extracts from Postimees Daily. All documents in the corpus are labeled with both sentiment and rubric classes. There are nine rubrics: Opinion, Estonia, Life, Comments-Life, Comments-Estonia, Crime, Culture, Sports, and Abroad. The four sentiment labels include Positive, Negative, Neutral, and Ambiguous. We split the data into 70/10/20 training, development and test sets, stratified over both rubric and sentiment analysis. Table 4 and Table 5 show the statistics about the sentiment and rubric view of the classification dataset respectively.

4 Pretraining EstBERT

The EstBERT model was pretrained on the architecture identical to the BERT_{Base} model with 12 transformer blocks with 768 hidden units each and 110M trainable parameters. It was pretrained on the Masked Language Modeling (MLM) and the Next Sentence Prediction (NSP) tasks as described by Devlin et al. (2019). In MLM, the probability of correctly predicting the randomly masked tokens is maximised. Because in the

	Train	Dev	Test	Total
Positive	612	87	175	874
Negative	1347	191	385	1923
Neutral	505	74	142	721
Ambiguous	385	55	110	550
Total	2849	407	812	4068

Table 4: Sentiment label statistics of the Estonian Valence corpus.

	Train	Dev	Test	Total
Opinion	676	96	192	964
Estonia	289	41	83	413
Life	364	52	101	517
Comments-Life	354	50	102	506
Comments-Estonia	351	50	100	501
Crime	146	21	42	209
Culture	182	27	51	260
Sports	269	39	77	385
Abroad	218	31	64	313
Total	2849	407	812	4068

Table 5: Rubric label statistics of the Estonian Valence corpus.

transformer architecture, the model can simultaneously see both the left and the right context of a masked word, optimizing the MLM gives the model a bidirectional understanding of a sentence, as opposed to only the left or right context provided by recurrent language models. The NSP involves optimizing a binary classification task to predict whether the two sequences in the input follow each other in the original text or not, where half of the time, the second sequence is the correct next sentence and the other half of the time the two sequences are unrelated. The models were trained on four NVIDIA Tesla V100 GPUs across two nodes of the High-performance Computing Center at the University of Tartu (University of Tartu, 2018).

The model was first trained with the sequence length of 128. Then we evaluated the checkpoints generated during pretraining on the tasks described in section 3 to choose the final model with that sequence length. Finally, the chosen model was used as a starting point for training the longer model with 512 sequence length. Thus, as a result of pretraining, two EstBERT models, one with maximum sequence length 128 and the other with maximum sequence length 512, were obtained. The following subsections describe these three steps in more detail.

4.1 Pretraining with Sequence Length 128

The model with the sequence length of 128 was pretrained for two phases, both for 900K steps. Although the number of training steps was chosen following Vir-

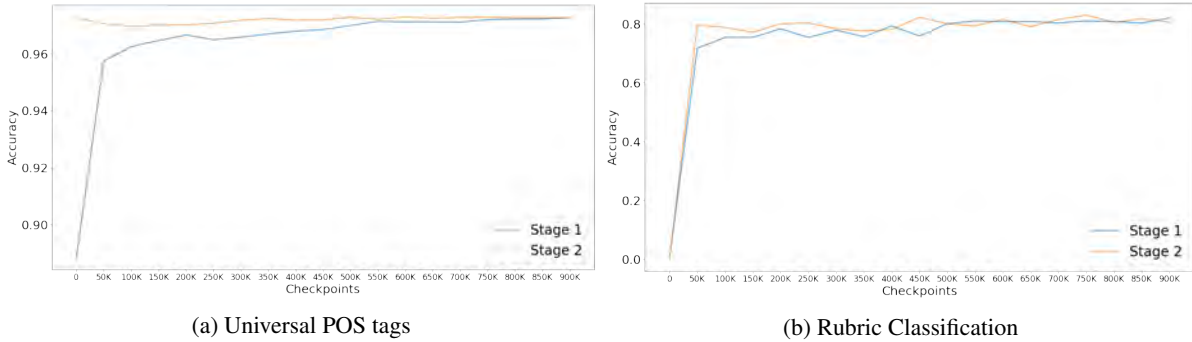


Figure 1: The validation performance on POS tagging and text classification tasks after every 50K checkpoints.

train_batch_size	32
max_seq_length	128
max_predictions_per_seq	20
num_train_steps	900000
num_warmup_steps	9000
learning_rate	1e-4
save_checkpoints_step	50000

Table 6: Hyperparameters used in the first two pretraining phases with the sequence length 128.

tanen et al. (2019), coincidentally this (900K steps) was the maximum number of steps we could fit in the given GPU time limit of 8 days. Therefore, the model was trained in two phases, each having 900K steps. A checkpoint was saved to the disk after every 50K steps. While the first phase of pretraining started from scratch with randomly initialised parameters, the second phase of training was initialised from the first phase’s last checkpoint. Since the GPU memory availability was a major issue, the batch size was kept at 32 to avoid the tensors going beyond the allowed GPU memory size. The BERT_{Base} uses Adam optimiser with weight decay. For EstBERT, the same optimiser was used with warmup over the first 1% of steps (9000) to a peak learning rate of 1e-4. The relevant hyperparameters are shown in Table 6. The pretraining process took around 192 hours for each phase.

4.2 Pretraining Validation

During pretraining, a checkpoint was saved after every 50K steps for later evaluation. To monitor the pretraining process, we evaluated the performance of MLM, NSP, and the evaluation tasks described in section 3 on all these checkpoints.

For POS and morphological tagging, and named entity recognition, we finetuned EstBERT using scripts from HuggingFace transformers library.⁷ A single randomly initialised fully connected classifier layer was trained on top of the token representations of the last hidden layer of the EstBERT model. All hyperparam-

⁷https://github.com/huggingface/transformers/blob/master/examples/token-classification/run_ner.py

eters were kept at their default values, which involves training for three epochs, using the learning rate of 5e-5 and batch size of 8. For the rubric and sentiment classification tasks, we adapted the classifier training scripts available in google’s BERT repository.⁸ The input to the single fully-connected classifier layer is the last hidden representation of the first token [CLS] in the input sequence. Here again, the classifier layer was initialised randomly and the default values for hyperparameters were used: training for three epochs with the learning rate 5e-5 and batch size 32. In all tasks, both the task-specific classification layer as well as the EstBERT parameters were finetuned.

The validation results of the masked language model, next sentence prediction accuracy, and all the evaluation tasks for all the eighteen checkpoints from stage one and other eighteen models from stage two were compared to pick the best model. The examples of validation curves for the UPOS tagging and the rubric classification tasks are shown in Figure 1. Although the checkpoint validation results from both phases showed more or less steady improvement with the increase of the number of steps trained, we observed that the checkpoint at 750K steps from phase two performs slightly better on all tasks than the rest of the checkpoints. Thus, this checkpoint was chosen as a final model with sequence length 128.

train_batch_size	16
max_seq_length	512
max_predictions_per_seq	77
num_train_steps	600000
num_warmup_steps	6000
learning_rate	1e-4
save_checkpoints_step	50000

Table 7: Hyperparameters used to pretrain the EstBERT model with the sequence length 512.

⁸https://github.com/google-research/bert/blob/master/run_classifier.py/

Model	Seq = 128			Seq = 512		
	UPOS	XPOS	Morph	UPOS	XPOS	Morph
EstBERT	<u>97.89</u>	98.40	96.93	97.84	<u>98.43</u>	96.80
WikiBERT-et	97.78	98.36	96.71	97.76	98.35	96.67
mBERT	97.42	98.06	96.24	97.43	98.13	96.13
XLM-RoBERTa	97.78	98.36	96.53	97.80	98.40	96.69

Table 8: POS and morphological tagging accuracy on the Estonian UD test set. The highest scores in each column are in **bold**. The highest overall score of each task is underlined.

4.3 Pretraining with Sequence Length 512

The starting point for training the model with a sequence length of 512 was the final model chosen for the sequence length 128. The longer model was trained further up to 600K steps. The batch size was reduced to 16 as the size of the tensors would be larger for the sequence length 512 compared to 128. The hyperparameters used to train the longer model are shown in Table 7. During training, checkpoints were again saved after every 50K steps, and these were evaluated on all evaluation tasks as previously explained in Section 4.2. Based on these evaluations, the last checkpoint obtained after the 600K steps was chosen as the final model with 512 sequence length.

5 Results

The next subsections present the results obtained with the final EstBERT models with both sequence lengths on the tasks described in section 3. We follow the same setup of Kittask et al. (2020) to enable direct comparison with the multilingual models. Some additional steps were taken to prepare the Estonian Valence corpus. First, all duplicate items, 17 in total, were removed. Also, all items with the Ambiguous label were removed as retaining them has been shown to lower the classification accuracy (Pajupuu et al., 2016). The same preprocessing was also applied in evaluating the multilingual BERT models for Estonian (Kittask et al., 2020).

For finetuning, we used the same scripts from the HuggingFace transformers repository that were used for the pretraining validation in section 4.2. The same scripts were also used to evaluate the multilingual models by Kittask et al. (2020). For each task, the learning rate of the AdamW optimiser and the batch size was tuned on the development set. The learning rate grid values were (5e-5, 3e-5, 1e-5, 5e-6, 3e-6) and the batch size grid values were (8, 16). The best model was found on the development set by using early stopping with the patience of 10 epochs.

We compare the results of EstBERT with the multilingual BERT models’ results from Kittask et al. (2020) and the WikiBERT model trained on the Estonian Wikipedia (Pyysalo et al., 2020). WikiBERT-et model was finetuned using the same setup described above.

Model	Seq = 128		Seq = 512	
	Rubr.	Sent.	Rubr.	Sent.
EstBERT	<u>81.70</u>	74.36	80.96	74.50
WikiBERT-et	72.72	68.09	71.13	69.37
mBERT	75.67	70.23	74.94	69.52
XLM-RoBERTa	80.34	74.50	78.62	<u>76.07</u>

Table 9: Rubric (Rubr.) and sentiment (Sent.) classification accuracy. The highest scores in each column are in **bold**. The highest overall score of each task is underlined.

5.1 POS and Morphological Tagging

The POS and morphological tagging results are summarised in Table 8 that shows the accuracy for universal POS tags (UPOS), language-specific POS tags (XPOS), and morphological features. The language-specific EstBERT outperforms all other models although the difference with the XLM-RoBERTa—the best-performing multilingual model—and the WikiBERT-et are quite small.

Similar to multilingual results, using longer sequence length on this task with the EstBERT model does not seem beneficial as the accuracy slightly increases only for XPOS tags but not for others. Overall, as the performances on these tasks are already very high, the absolute performance gains cannot be large. EstBERT obtains consistent improvements over mBERT, with the relative error reduction with both models on all tasks falling between 16-18%. The relative error reduction of the EstBERT compared to XLM-RoBERTa is smaller, in the range of 2-5%. The highest reduction of error of EstBERT compared to XLM-RoBERTa can be observed on the morphological tagging task with the shorter model where the relative error reduction is 12%. The WikiBERT-et model achieves almost identical results to XLM-RoBERTa with both sequence lengths.

5.2 Rubric and Sentiment Classification

The rubric and sentiment classification results are shown in Table 9. EstBERT outperforms mBERT and WikiBERT-et on both tasks by a large margin, but XLM-RoBERTa exceeds EstBERT on sentiment clas-

Model	Seq = 128			Seq = 512		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
EstBERT	89.10	91.15	90.11	88.35	89.74	89.04
WikiBERT-et	89.86	90.83	90.34	88.31	90.96	89.61
mBERT	85.88	87.09	86.51	88.47	88.28	88.37
XLm-RoBERTa	87.55	91.19	89.34	87.50	90.76	89.10

Table 10: NER tagging results. Upper section shows the comparison between different models. The highest scores in each column are in **bold**. The highest overall score of each measure is underlined.

Entity	EstBERT			XLm-RoBERTa			WikiBERT-et		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
PER	94.80	95.77	95.28	96.42	94.45	95.43	94.87	94.45	94.66
ORG	78.38	82.64	80.45	75.48	82.12	78.66	82.25	81.61	81.92
LOC	89.94	91.38	90.66	86.06	93.99	89.85	88.89	92.99	90.89

Table 11: The entity-based scores for the EstBERT, XLm-RoBERTa and the WikiBERT-et models. The best scores are in **bold**.

sification. The difference between the two accuracy scores is relatively small when the model with sequence length 128 is used, but it increases when the longer sequence length is used.

Like XLm-RoBERTa, the EstBERT model with a shorter sequence length is somewhat better on rubric classification, and the opposite is true for sentiment classification. Overall, the differences between the EstBERT models’ performances with both sequence lengths are again relatively small.

5.3 Named Entity Recognition

Table 10 shows the entity-based precision, recall, and F-score of the named entity recognition task. WikiBERT-et model is the best model in this task, obtaining the highest F1-score with both the short and long models and the overall highest F1-score with the short model. XLm-RoBERTa achieves the highest recall in the short model category but remains below the EstBERT in terms of the F1-score. EstBERT, WikiBERT-et and XLm-RoBERTa all benefit from using the smaller sequence length rather than longer, while mBERT shows the opposite behavior.

Table 11 shows the fine-grained scores of each entity type for both the EstBERT, XLm-RoBERTa and the WikiBERT-et shorter model. In alignment with the previous results in Estonian NER (Tkachenko et al., 2013), the prediction of the person entities is the most accurate while the organisation names are the most difficult to predict. The WikiBERT-et is the best on the two most difficult entities ORG and LOC, while the EstBERT model is better than XLm-RoBERTa on these two entities. The WikiBERT-et is notably the best on the most challenging organisation entity, improving the precision over the EstBERT model for almost 4% and over the XLm-RoBERTa for almost 7%, with a considerably smaller loss in recall. One reason for the superiority of

the WikiBERT-et model might lie in the fact that the Wikipedia dataset used to train the WikiBERT-et model probably contains a much higher proportion of organisation names. Although the datasets used to train the other two models also contain the Estonian Wikipedia dataset, it has been diluted in other languages (in case of XLm-RoBERTa) or genres (in case of EstBERT). However, this is just a hypothesis at the moment that has to be studied more in further work.

5.4 Dependency Parsing

Additionally, we also evaluated both the EstBERT, WikiBERT-et and the XLm-RoBERTa models on the Estonian dependency parsing task. The data used in these experiments is the Estonian UDv2.5 described in section 3.1. We trained the parser available in the spaCy Nightly version⁹ that also supports transformers. The models were trained with a batch size of 32 and for a maximum of 20K steps, stopping early when the development set performance did not improve for 1600 steps. The parser was trained jointly with a tagger component that predicted the concatenation of POS tags and morphological features. During training, the model was supplied with the gold sentence and token segmentations. During evaluation, the sentence segmentation and tokenisation was done with the out-of-the-box spaCy tokeniser.

The dependency parsing results are in Table 12. In addition to the transformer-based EstBERT and XLm-RoBERTa models, the right-most section also displays the Stanza parser (Qi et al., 2020), trained on the same Estonian UDv2.5 corpus, obtained from the Stanza web page.¹⁰ We add a non-transformer based baseline for this task because dependency parsing was not evaluated

⁹<https://pypi.org/project/spacy-nightly/>

¹⁰<https://stanfordnlp.github.io/stanza/models.html>

Model DepRel	Support	EstBERT			XLM-RoBERTa			WikiBERT-et			Stanza		
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
UAS		86.07	<u>87.34</u>	<u>86.70</u>	88.02	89.32	88.66	85.97	87.24	86.60	<u>86.69</u>	86.68	86.69
LAS		83.32	<u>84.56</u>	<u>83.94</u>	85.60	86.87	86.23	83.06	84.29	83.67	<u>83.63</u>	83.63	83.63
nmod	4328	81.40	<u>85.65</u>	<u>83.47</u>	83.73	88.49	86.05	80.86	85.42	83.08	<u>82.53</u>	84.27	83.39
obl	4198	<u>80.99</u>	<u>79.78</u>	<u>80.38</u>	83.65	82.66	83.15	80.81	79.47	80.13	79.61	77.78	78.68
advmod	3938	<u>78.01</u>	<u>79.02</u>	<u>78.52</u>	80.43	80.45	80.44	78.08	77.96	78.02	<u>78.93</u>	78.11	<u>78.52</u>
root	3214	<u>90.82</u>	89.61	<u>90.21</u>	91.88	91.91	91.90	90.32	<u>89.73</u>	90.03	90.18	87.46	88.80
nsubj	2682	<u>92.05</u>	<u>93.25</u>	<u>92.65</u>	93.54	94.52	94.03	90.40	92.69	91.53	90.67	89.90	90.28
conj	2476	76.90	<u>78.51</u>	<u>77.70</u>	81.72	83.44	82.57	<u>78.28</u>	78.31	<u>78.30</u>	76.41	<u>78.76</u>	77.57
obj	2437	<u>86.91</u>	<u>88.80</u>	<u>87.84</u>	88.62	90.73	89.66	86.36	87.57	86.96	83.51	84.78	84.14
amod	2411	<u>80.02</u>	<u>84.20</u>	<u>82.05</u>	82.90	86.64	84.73	80.12	83.91	81.97	91.93	89.26	90.57
cc	2029	<u>91.26</u>	<u>90.09</u>	<u>90.67</u>	92.57	91.47	92.02	90.75	89.50	90.12	<u>89.97</u>	88.42	89.19
aux	1372	<u>95.36</u>	<u>95.85</u>	<u>95.60</u>	95.43	95.99	95.71	94.79	95.48	95.13	89.93	95.04	92.42
mark	1277	<u>90.14</u>	<u>90.92</u>	<u>90.53</u>	92.75	93.19	92.97	89.25	89.74	89.50	88.35	89.12	88.73
cop	1202	<u>84.75</u>	<u>87.35</u>	<u>86.03</u>	85.48	87.69	86.57	84.29	87.02	85.63	81.43	86.11	83.70
acl	1063	84.98	<u>85.14</u>	<u>85.06</u>	86.88	87.86	87.37	<u>86.67</u>	85.04	<u>85.85</u>	86.36	80.43	83.29
nsubj:cop	1054	<u>79.98</u>	<u>82.64</u>	<u>81.29</u>	81.16	85.01	83.04	79.34	<u>82.73</u>	81.00	77.78	79.70	78.73
case	908	<u>92.42</u>	<u>92.62</u>	<u>92.52</u>	93.52	93.72	93.62	91.32	<u>92.73</u>	92.02	89.13	91.19	90.15
advcl	857	<u>67.18</u>	65.93	<u>66.55</u>	73.46	71.06	72.24	66.55	<u>66.39</u>	66.47	67.12	63.36	65.19
det	808	83.88	<u>85.02</u>	<u>84.45</u>	87.89	87.13	87.51	82.95	84.28	83.61	80.80	82.80	81.78
parataxis	725	52.96	49.38	51.11	57.50	50.76	53.92	55.59	48.69	51.91	65.45	59.31	62.23
xcomp	641	<u>85.21</u>	<u>87.21</u>	<u>86.20</u>	88.06	88.61	88.34	84.11	86.74	85.41	83.78	83.00	83.39
flat	633	81.44	85.94	83.63	86.64	91.15	88.84	80.09	86.41	83.13	88.60	92.10	90.32
nummod	555	62.88	77.84	69.57	62.00	80.54	70.06	63.12	78.02	69.78	85.53	85.23	85.38
compound:prt	481	<u>86.10</u>	92.72	89.29	88.20	94.80	91.38	85.99	<u>93.14</u>	<u>89.42</u>	85.52	89.60	87.51
appos	376	69.07	71.28	70.16	74.45	80.59	77.39	64.55	<u>73.14</u>	68.58	<u>69.47</u>	72.61	71.00
ccomp	344	<u>82.56</u>	82.56	82.56	87.03	87.79	87.41	80.44	<u>84.88</u>	<u>82.60</u>	81.87	78.78	80.30
acl:relcl	315	80.67	83.49	82.06	79.00	80.00	79.50	79.30	79.05	79.17	61.32	82.54	70.37
csubj:cop	121	80.47	85.12	82.73	75.74	85.12	80.16	79.23	85.12	82.07	72.79	88.43	79.85
csubj	108	81.51	89.81	85.46	80.83	89.81	85.09	84.26	84.26	84.26	84.91	83.33	84.11
discourse	47	37.14	55.32	44.44	36.92	51.06	42.86	34.33	48.94	40.35	81.25	55.32	65.82
orphan	44	20.83	11.36	14.71	37.93	25.00	30.14	20.00	18.18	19.05	45.00	20.45	28.12
compound	43	88.10	86.05	87.06	83.33	81.40	82.35	92.11	81.40	86.42	88.64	90.70	89.66
cc:preconj	39	66.67	71.79	69.14	67.57	64.10	65.79	62.79	69.23	65.85	70.27	66.67	68.42
flat:foreign	37	<u>76.19</u>	43.24	<u>55.17</u>	87.50	56.76	68.85	43.75	18.92	26.42	65.38	45.95	53.97
fixed	31	64.71	70.97	67.69	64.86	77.42	70.59	57.50	<u>74.19</u>	64.79	75.86	70.97	73.33
vocative	9	22.22	22.22	22.22	28.57	22.22	25.00	5.56	11.11	7.41	30.77	44.44	36.36
goeswith	8	100.00	12.50	22.22	25.00	12.50	16.67	33.33	25.00	28.57	0.00	0.00	0.00
dep	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
list	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 12: Dependency parsing results. The best scores over all models are in **bold**. The best scores comparing the EstBERT, WikiBERT-et and the Stanza models are underlined.

by Kittask et al. (2020). Overall, the XLM-RoBERTa model performs the best, both in terms of the UAS and LAS metrics and the individual dependency relations. This is especially true for dependency relations with larger support in the test set. Although in terms of the UAS and LAS, the EstBERT, WikiBERT-et and Stanza models seem to perform similarly, a closer look into the scores of the individual dependency relations reveals that in most cases, especially with relations of larger support, the EstBERT model performs the best. There are few dependency relations where the Stanza system’s predictions are considerably more accurate than the BERT-based models, the most notable of them being the adjectival modifier (amod) and the numerical modifier (nummod). Further analyses are needed to gain more insight into these results.

6 Discussion

This objective of this paper was to describe the process of pretraining the language-specific BERT model for Estonian and to compare its performance with the multilingual BERT models as well as with the smaller Estonian WikiBERT model on several NLP tasks. Overall, the pretrained EstBERT was better than the best multilingual XLM-RoBERTa model on five tasks out of seven: UPOS, XPOS, and morphological tagging, rubric classification, and NER. Only in the sentiment classification and dependency parsing tasks, the XLM-RoBERTa was better. Compared to WikiBERT-et, the EstBERT model was better on six tasks out of seven—the WikiBERT-et model was superior only in the NER task, predicting ORG entities considerably better than any other model. We did not observe any consistent difference between the models of different sequence lengths, although the model with the sequence length

512 was trained longer. It is possible that the shorter model was already trained long enough, and the subsequent training of the longer model did not add any effect in that respect, aside from the fact that it can accept longer input sequences.

One crucial aspect of this work was obtaining a large-enough corpus for pretraining the model. We used the Estonian National Corpus 2017, which was the largest corpus available at the time. A newer and larger version of this corpus—the Estonian National Corpus 2019 (Kallas and Koppel, 2019)—has become available meanwhile. There are also few other resources, such as the Estonian part of the CoNLL 2017 raw data (Ginter et al., 2017) and the Oscar Crawl, which probably partially overlap with each other and with the Estonian National Corpus. Still, these corpora would potentially provide additional data that was currently not used.

Another challenge was related to finding annotated datasets for downstream tasks. While the Estonian UD dataset provides annotations to the common dependency parsing pipeline tasks, datasets for other, especially semantic NLP tasks, are scarce. We adopted the Estonian Valence corpus for two-way text classification. However, the labels of this corpus are semi-automatically derived from user ratings, and thus the quality of these annotations cannot be guaranteed. An Estonian coreference dataset with some simple baseline results in nominal coreference resolution has recently become available (Barbu et al., 2020), which gives further opportunities to test out the EstBERT model in future work.

When preprocessing the data and pretraining the model, we mostly followed the process of training the FinBERT model for Finnish (Virtanen et al., 2019). We also decided to truecase our corpus to reduce the number of capitalised words in the vocabulary. The EstBERT model itself was also pretrained on the truecased corpus. However, when training the task-based models for evaluation, the EstBERT was finetuned on the cased datasets. Thus, truecasing the datasets before finetuning might have a positive effect on the results. In order to verify this, the EstBERT-based task-specific models should be finetuned using the truecased annotated datasets as input, and compared with the results reported in this paper.

Although we did see some improvements with EstBERT compared to XLM-RoBERTa on the smaller model for the NER task, the differences in scores were generally relatively small. However, we have observed that the annotations of this NER dataset are occasionally erroneous, containing, for instance, label sequences (I-PER, I-PER) instead of (B-PER, I-PER). We have also observed unlabelled entities in the text. Thus, the small variations in the systems’ results might not be informative about the systems themselves but can instead stem from the noise in the data. Although these annotation errors have been noticeable enough,

the magnitude of these errors has not been quantified.

The differences between the EstBERT and the XLM-RoBERTa model were, in most cases, relatively small. In previous experiments with several multilingual BERT models on the same Estonian tasks (Kittask et al., 2020), the XLM-RoBERTa proved to be the best multilingual model. This suggests that one option to obtain an even better Estonian language-specific model would be to train an Estonian-specific RoBERTa by initializing the model with the parameters of the XLM-RoBERTa. Considering that the multilingual RoBERTa already performs very well on Estonian tasks, finetuning it with more Estonian data would hopefully bias it even more to the Estonian language while also maintaining the gains obtained from multilingualism.

7 Conclusion

We presented EstBERT, the largest BERT model pretrained specifically on the Estonian language. While several existing multilingual BERT models include Estonian, the only language-specific Estonian BERT model available until now has been trained on the relatively small Wikipedia data. In order to pretrain the EstBERT model, we used the largest Estonian text corpus available at the time. The EstBERT model was put to the test by finetuning it for several tasks, including POS and morphological annotations, dependency parsing, named entity recognition, and text classification. On five tasks out of seven, the classifiers based on EstBERT achieved better performance than the models based on multilingual BERT models, although in several cases, the gap with the best-performing multilingual XLM-RoBERTa was relatively small. These results suggest that training a RoBERTa model for Estonian, initialised with the multilingual model’s parameters, might be beneficial. On six tasks out of seven, the models based on EstBERT achieved better results than the Estonian BERT model trained on Wikipedia, suggesting that using more textual data for pretraining leads to a more generalisable model.

References

- Eduard Barbu, Kadri Muischnek, and Linda Freienthal. 2020. [A Study in Estonian Pronominal Coreference Resolution](#). In *Volume 328: Human Language Technologies – The Baltic Perspective*, *Frontiers in Artificial Intelligence and Applications*, pages 3–10.
- José Cañete, Gabriel Chaperon, Rodrigo Fuentes, and Jorge Pérez. 2020. Spanish pre-trained BERT model and evaluation data. *PMLADC at ICLR*.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Unsupervised Cross-Lingual Representation Learning at Scale](#). *arXiv preprint arXiv:1911.02116*.

- Alexis Conneau and Guillaume Lample. 2019. [Cross-Lingual Language Model Pretraining](#). In *Advances in Neural Information Processing Systems*, pages 7059–7069.
- Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. [Pre-Training with Whole Word Masking for Chinese BERT](#). *arXiv preprint arXiv:1906.08101*.
- Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. [RobBERT: a Dutch RoBERTa-based Language Model](#). *arXiv preprint arXiv:2001.06286*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. [CoNLL 2017 Shared Task - Automatically Annotated Raw Texts and Word Embeddings](#). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- J. Kallas and K. Koppel. 2018. [Estonian National Corpus 2017](#). Center of Estonian Language Resources.
- J. Kallas and K. Koppel. 2019. [Estonian National Corpus 2019](#). Center of Estonian Language Resources.
- Claudia Kittask, Kirill Milintsevich, and Kairit Sirts. 2020. [Evaluating multilingual BERT for Estonian](#). In *Volume 328: Human Language Technologies – The Baltic Perspective*, *Frontiers in Artificial Intelligence and Applications*, pages 19–26.
- Yuri Kuratov and Mikhail Arkhipov. 2019. [Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language](#). *Computational Linguistics and Intellectual Technologies*, (18):333–339.
- Sven Laur, Siim Orasmaa, Dage Särg, and Paul Tamm. 2020. [EstNLTK 1.6: Remastered Estonian NLP Pipeline](#). In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 7154–7162.
- Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2020. [FlauBERT: Unsupervised Language Model Pre-training for French](#). In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490.
- Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. [CamemBERT: a Tasty French Language Model](#). *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219.
- Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. [What the \[MASK\]? Making Sense of Language-Specific BERT Models](#). *arXiv preprint arXiv:2003.02912*.
- Hille Pajupuu, Rene Altrov, and Jaan Pajupuu. 2016. Identifying Polarity in Different Text Types. *Folklore*, 64.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep Contextualized Word Representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.
- Sampo Pyysalo, Jenna Kanerva, Antti Virtanen, and Filip Ginter. 2020. [Wikibert models: deep transfer learning for many languages](#). *arXiv preprint arXiv:2006.01538*.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. [Stanza: A python natural language processing toolkit for many human languages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. [DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter](#). *arXiv preprint arXiv:1910.01108*.
- Alexander Tkachenko, Timo Petmanson, and Sven Laur. 2013. [Named Entity Recognition in Estonian](#). In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 78–83.
- University of Tartu. 2018. [UT Rocket](#). share.neic.no.
- Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. [Multilingual is not enough: BERT for Finnish](#). *arXiv preprint arXiv:1912.07076*.
- Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. [BERTje: A Dutch BERT Model](#). *arXiv preprint arXiv:1912.09582*.

Operationalizing a National Digital Library: The Case for a Norwegian Transformer Model

Per E Kummervold

per.kummervold@nb.no

Javier de la Rosa

javier.rosa@nb.no

Freddy Wetjen

freddy.wetjen@nb.no

Svein Arne Brygfeld

svein.brygfeld@nb.no

The National Library of Norway
Mo i Rana, Norway

Abstract

In this work, we show the process of building a large-scale training set from digital and digitized collections at a national library. The resulting Bidirectional Encoder Representations from Transformers (BERT)-based language model for Norwegian outperforms multilingual BERT (mBERT) models in several token and sequence classification tasks for both Norwegian Bokmål and Norwegian Nynorsk. Our model also improves the mBERT performance for other languages present in the corpus such as English, Swedish, and Danish. For languages not included in the corpus, the weights degrade moderately while keeping strong multilingual properties. Therefore, we show that building high-quality models within a memory institution using somewhat noisy optical character recognition (OCR) content is feasible, and we hope to pave the way for other memory institutions to follow.

1 Introduction

Modern natural language processing (NLP) models pose a challenge due to the massive size of the training data they require to perform well. For resource-rich languages such as Chinese, English, French, and Spanish, collections of texts from open sources such as Wikipedia (2021a), variations of Common Crawl data (2021), and other open-source corpora such as the BooksCorpus (Zhu et al., 2015) are generally used. When researchers at Google released their Bidirectional Encoder Representations from Transformers (BERT) model, they trained it on a huge corpus of 16GB of uncompressed text (3,300M words)

(Devlin et al., 2019). Later research has shown that the corpus size might have even been too small, and when Facebook released its Robustly Optimized BERT (RoBERTa), it showed a considerable gain in performance by increasing the corpus to 160GB (Liu et al., 2019).

Norwegian is spoken by just 5 million people worldwide. The reference publication *Ethnologue* lists the 200 most commonly spoken native languages, and it places Norwegian as number 171. The Norwegian language has two different varieties, both equally recognized as written languages: Bokmål and Nynorsk. The number of Wikipedia pages written in a certain language is often used to measure its prevalence, and in this regard, Norwegian Bokmål ranges as number 23 and Nynorsk as number 55. However, there exist more than 100 times as many English Wikipedia pages as there are Norwegian Wikipedia pages (2021b). When it comes to building large text corpora, Norwegian is considered a minor language, with scarce textual resources. So far, it has been hard to train well-performing transformer-based models for such languages.

As a governmental entity, the National Library of Norway (NLN) established in 2006 a mass digitization program for its collections. The Language Bank, an organizational unit within the NLN, provides text collections and curated corpora to the scholarly community (Språkbanken, 2021). Due to copyright restrictions, the publicly available Norwegian corpus consists mainly of Wikipedia pages and online newspapers, and it is around 5GB (818M words) in size (see Table 1). However, in this study, by adding multiple sources only accessible from the NLN, we were able to increase that size up to 109GB (18,438M words) of raw, deduplicated text. While such initiatives may produce

textual data that can be used for the large-scale pre-training of transformer-based models, relying on text derived from optical character recognition (OCR)-based pipelines introduces new challenges related to the format, scale, and quality of the necessary data. On these grounds, this work describes the effort to build a pre-training corpus and to use it to train a BERT-based language model for Norwegian.

1.1 Previous Work

Before the advent of transformer-based models, non-contextual word and document embeddings were the most prominent technology used to approach general NLP tasks. In the Nordic region, the Language Technology Group at the University of Oslo, as part of the joint Nordic Language Processing Laboratory, collected a series of monolingual resources for many languages, with a special emphasis on Norwegian (Kutuzov et al., 2017). Based on these resources, they trained and released collections of dense vectors using word2vec and fastText (both with continuous skip-gram and continuous bag-of-words architectures) Mikolov et al. 2013; Bojanowski et al. 2017, and even using an Embeddings from Language Models (ELMo)-based model with contextual capabilities (Peters et al., 2018). Shortly thereafter, Devlin et al. (2019) introduced the foundational work on the monolingual English BERT model, which would later be extended to support 104 different languages including Norwegian Bokmål and Norwegian Nynorsk, Swedish, and Danish. The main data source used was Wikipedia (2021a). In terms of Norwegian, this amounted to around 0.9GB of uncompressed text (140M words) for Bokmål and 0.2GB (32M words) for Nynorsk (2021b). While it is generally agreed that language models acquire better language capabilities by pre-training with multiple languages (Pires et al., 2019; Wu and Dredze, 2020), there is a strong indication that this amount of data might have been insufficient for the multilingual BERT (mBERT) model to learn high-quality representations of Norwegian at a level comparable to, for instance, monolingual English models (Pires et al., 2019).

In the area of monolingual models, the Danish company BotXO trained BERT-based models for a few of the Nordic languages using corpora of various sizes. Their repository (BotXO Ltd., 2021) lists models trained mainly on Common Crawl

data for Norwegian (5GB), Danish (9.5GB), and Swedish (24.7GB). Unfortunately, we were unable to make the Norwegian models work, as they seem to be no longer updated. Similarly, the KBLab at the National Library of Sweden trained and released a BERT-based model and an A Lite BERT (ALBERT) model, both trained on approximately 20GB of raw text from a variety of sources such as books, news articles, government publications, Swedish Wikipedia, and internet forums (Malmsten et al., 2020). They claimed significantly better performance than both the mBERT and the Swedish model by BotXO for the tasks they evaluated.

At the same of the release of our model, the Language Technology Group at the University of Oslo released a monolingual BERT-based model for Norwegian named NorBERT. It was trained on around 5GB of data from Wikipedia and the Norsk aviskorpus (2019). We were unable to get sensible results when finetuning version 1.0 of their model. However, they released a second version shortly thereafter (1.1) fixing some errors (Language Technology Group at the University of Oslo, 2021a). We have therefore included the evaluation results of this second version of the model in our benchmarking. They have also evaluated their and our model themselves (Kutuzov et al., 2021) with consistent results.

2 Building a Colossal Norwegian Corpus

As the main Norwegian memory institution, the NLN has the obligation to preserve and give access to all published information in Norway. A large amount of the traditional collection is now available in digital format. As part of the current legal deposit, many born-digital documents are also available as digital documents in the collection. The texts in the NLN collection span hundreds of years and exhibit varied uses of texts in society. All kinds of historical written materials can be found in the collections, although we found that the most relevant resources for building an appropriate corpus for NLP were books, magazines, journals, and newspapers (see Table 1). As a consequence, the resulting corpus reflects the variation in the use of the Norwegian written language, both historically and socially.

Texts in the NLN have been subject to a large digitization operation in which digital copies were created for long-term preservation. The NLN em-

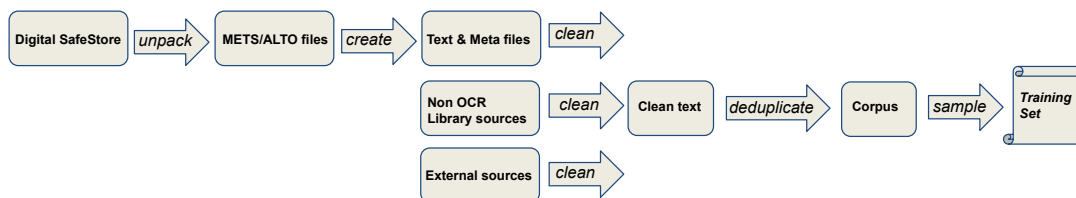


Figure 1: The general corpus-building process.

employs METS/ALTO¹ as the preferred format for storing digital copies. As the digitized part of the collection conforms to standard preservation library practices, the format in which the texts are stored is not suitable for direct text processing; thus, they needed to be pre-processed and manipulated for use as plain text. One major challenge was the variation in the OCR quality, which varied both over time and between the types of materials digitized. This limited the number of usable resources and introduced some artifacts that affected the correctness of the textual data.

The basic inclusion criterion for our corpus was that as long as it was possible for a human to infer the meaning from the text, it should be included. However, the amount of text involved in building the model meant that this needed to be determined automatically. The METS/ALTO files contain information from the OCR process regarding the confidence of every word (from 0 for no confidence to 1 for certainty), so we used this assessment to calculate the average confidence for paragraphs and pages. Setting the minimum paragraph confidence to 0.8 and the minimum page confidence to 0.9 allowed us to filter out a significant part of the text with the lowest quality. We also noticed that in the period of digitization from the beginning of 2006 until the end of 2008, the quality of the OCR was low and the estimated confidence values were too optimistic. We ended up excluding all text scanned in this period.

To further filter out erroneous textual information, we calculated the number of words in the documents and averaged the number of words per paragraph. Establishing a threshold of at least 20 words per document and an average of 6 words per paragraph, we could filter out text sources that had little value for training, such as cartoons and picture books. We estimated the language composition using various methods, including metadata

¹Metadata Encoding and Transmission Schema and Analyzed Layout and Text Object (Library of Congress, 2020, 2016)

tags in the collection and counting the frequency of words of certain types (e.g., personal pronouns). Our estimate is that 83% of the text is in Norwegian Bokmål and 12% is in Nynorsk. Close to 4% of the texts are written in English, and the 1% left is a mixture of Sami, Danish, Swedish, and a few traces from other languages.

The aforementioned process was carefully orchestrated, with data moving from preservation storage, through error correction and quality assessment, and ending up as text in the corpus. As shown in Figure 1, after filtering, OCR-scanned documents were added to the other digital sources. After this step, the data went through the cleaning process, in which we ensured the consistency of the text encoding and special characters used. In the deduplication stage, all duplicated paragraphs in the entire collection were removed. Finally, we drew out two pre-training-sets: one with a sequence length of 128 tokens, and one with a sequence length of 512 tokens.

3 Pre-training a Norwegian BERT model

In order to build our own pre-trained language model for Norwegian, we decided to use the original BERT architecture pre-trained with a masked-language model (MLM) objective, as published by Devlin et al. (2019). We evaluated the effect of changes in hyperparameters in terms of MLM performance and of the fine-tuning of the pre-trained models on various downstream tasks. All pre-training work was run on a v3-8 TPU (128GB) provided by the TPU Research Cloud, while the evaluation was done on in-house machines with a single NVIDIA Quadro RTX6000 (24GB).

Our goal was to build a solid model that would perform well on all types of Norwegian language tasks, ranging from old to modern text, and including texts that might be mixed with foreign languages like English. We therefore chose to initiate the model from the pre-trained mBERT weights (TensorFlow Hub, 2021). The mBERT

Sources	Period	Words (Millions)	Text (GB)
Books (OCR)	1814–2020	11,820	69.0
Newspaper Scans (OCR)	2015–2020	3,350	20.0
Parliament Documents ^a (OCR)	1814–2014	809	5.1
Common Crawl OSCAR	1991–2020	799	4.9
Online Bokmål Newspapers	1998–2019	678	4.0
Periodicals (OCR)	2010–2020	317	1.9
Newspaper Microfilms (OCR)	1961, 1971, 1981, 1998–2007	292	1.8
Bokmål Wikipedia	2001–2019	140	0.9
Public Reports ^b (OCR)	1814–2020	91	0.6
Legal Collections ^c	1814–2004	63	0.4
Online Nynorsk Newspapers	1998–2019	47	0.3
Nynorsk Wikipedia	2001–2019	32	0.2
Total (After Deduplication)		18,438	109.1

^aStortingsforhandlingene. ^bEvalueringsrapporter. ^cLovdata CD/DVD.

Table 1: The composition of the Colossal Norwegian Corpus.

model was trained on 104 languages, including both Norwegian varieties (Bokmål and Nynorsk). The model uses a 119,547-token vocabulary, and its pre-trained weights might also benefit from cross-lingual transfer. Our assumption is that using the mBERT weights for Norwegian should result in a better-performing model in comparison to starting with random weights. It might also keep some of its multilingual abilities, making it more robust when dealing with new words and texts containing fragments of other languages (Wu and Dredze, 2020).

3.1 Improving the Model Beyond mBERT

All subsequent training runs followed the findings by You et al. (2019), who showed that the pre-training of a BERT model could be improved by increasing the batch size but that, at the same time, an increase in the learning rate could lead to instability, especially when using the adaptive moment estimation (Adam) optimizer. When training on large batch sizes, You et al. suggested using their layer-wise adaptive moments base (LAMB) optimizer instead. We confirmed these results on our dataset when pre-training for 100,000 steps on a batch size of 2,048 sequences, which is very close to the optimum size for our v3-8 TPU (128GB) setup (see Figure 2).

The basic pre-training strategy was to use the largest possible batch size on our TPU and to increase the learning rate as long as it showed stability. An evaluation of the learning rate was done for 100,000 steps, but because we used decay, we

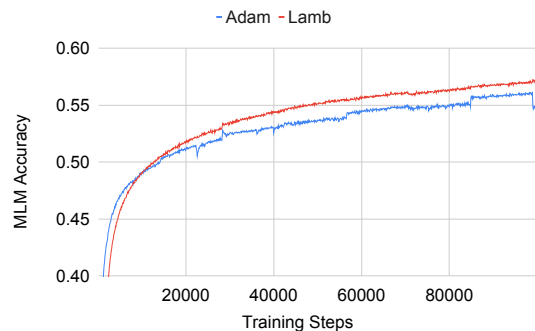


Figure 2: Comparison of Adam and LAMB optimizers (learning rate: 4e-4; batch size: 2,048).

expected the stability to be maintained even after this point. Devlin et al. (2019) trained for 128-length sequences for approximately 90% of the training examples, then trained for 512-length sequences for 10%. Due to memory limits on our TPUs, we needed to reduce the batch size (by a factor of approximately 7) for the 512 sequences in the pre-training data; we also increased the number of pre-training steps for the long sequences to resemble the same distribution of short and long sequences that were used in training the BERT model. To investigate the effect of this, we experimented with two different setups in our model (version A and version B). Both were initialized from the same mBERT weights and trained identically for the first 1,750,000 steps. In the last steps, version A followed the training schedule used in the BERT model where roughly 10% of the total training time was used on long sequences (step

3a) and then an additional step (3b) on shorter sequences. Version B reduced the training on short sequences and instead trained almost 30% of the time on long sequences. The setup was chosen for making the total training time roughly the same for both models (see Table 2).

4 Evaluation

While pre-trained language models also can be used for direct MLM-prediction and feature extractions, the most common use is to fine-tune it on a specific task. The base procedure for fine-tuning was described by Vaswani et al. (2017), and it consists of training for a small number of epochs (typically 4), with a warmup of around 10% of the training steps; subsequently, a linear decay to zero is used. Devlin et al. (2019) based their work on the same procedure and selected the best learning rate among $5e-5$, $3e-5$, and $2e-5$, according to the performance of the model on the validation set. The optimal learning rate and number of epochs mainly depend on the size of and variance in the training corpus, but they can also be affected by the properties of the pre-trained model. To get optimal performance out of a pre-trained model, the hyperparameters in the fine-tuning should be adapted. However, in this work, we are not primarily interested in optimization but in a comparison of the performance of our models against the mBERT model.

4.1 Token Classification

A common way to evaluate language models is by fine-tuning the models on token classification tasks such as named-entity recognition (NER) and part-of-speech (POS) tagging. For Norwegian, the Norwegian Dependency Treebank (NDT, Solberg et al., 2014) by the Språkbanken at the NLN and the Language Technology Group at the University of Oslo provide text that has been manually annotated with morphological features, syntactic functions, and hierarchical structures. The morphological annotation mainly follows the Oslo-Bergen tagger (Johannessen et al., 2012), and with a few exceptions, the syntactic analysis follows the Norwegian Reference Grammar (Faarlund et al., 1997). With the help of Schibsted Media Group, the same group recently published Norwegian Named Entities (NorNE) (Jørgensen et al., 2020), an extension of NDT that includes named-entity annotations for more than 300,000 tokens.

Moreover, with the goal of testing being the retaining or vanishing of the multilingual abilities of our model, we also considered NER datasets in both languages included in our corpus and in languages of which there is little to no evidence in our corpus. Specifically, we used CoNLL-2003 for English (Tjong Kim Sang and De Meulder, 2003), Webbnhyheter 2012 for Swedish (Gothenburg University Språkbanken, 2012), DaNE for Danish (Hvingelby et al., 2020), CoNLL-2002 for Spanish (Tjong Kim Sang, 2002), and FiNER for Finnish (Ruokolainen et al., 2019). While the number and specificity of the tag sets vary across datasets, rendering the comparison between languages useless, we could still compare the performance of our model against that of English-only and multilingual BERT models. We decided to leave out NER datasets built using automated or semi-automated annotations processes.

4.2 Sequence Classification

For sequence classification, we chose another commonly used task: sentiment classification. We used a version of the Norwegian Review Corpus (NoReC) (Øvrelid et al., 2020), a fine-grained sentiment dataset (Language Technology Group at the University of Oslo, 2021b) for Norwegian created by the Nordic Language Processing Laboratory. The fine-grained annotations in NoReC_{fine} were aggregated, and sentences with conflicting sentiments or no sentiment were removed. Moreover, we defined a second sequence-classification task to capture the idiosyncrasies and nuances of the Norwegian language. In this case, we generated a balanced corpus of 6,000 text speeches that had been spoken at the Norwegian Parliament (Storting) between 1998 and 2016 by members of the two major parties, Fremskrittspartiet and Sosialistisk Venstreparti (Lapponi et al., 2018). The dataset is annotated with the party the speaker was associated with at the time, and the source data was made publicly available by the Norwegian parliament. The classification task is to determine the political affiliation of the transcribed speech segment.

5 Results

To evaluate the performance of our model, we searched for the optimal set of fine-tuning hyperparameters for each downstream task by running a small grid search (see Table 3) on the mBERT

	Warmup	Step 1	Step 2	Version A		Version B
				Step 3a	Step 3b	Step 3
Steps	50k	700k	1M	1.2M	1.2M	2M
Batch Size	2760	2760	384	384	2760	384
Examples	138M	1,938M	384M	460M	3,312M	768M
Sequence Length	128	128	512	512	128	512
Learning Rate	0 \rightarrow 4e-4	4e-4	4e-4	4e-4 \rightarrow 2e-4	2e-4 \rightarrow 0	4e-4 \rightarrow 0

Table 2: Training schedule for our models.

model. The search space was the same for all tasks and included learning rates ranging from $2e-5$ to $5e-5$, with the number of training epochs being 3 or 4. We did the same for the warmup ratio and weight decay. The performance was generally best using a warmup ratio of 0.1 and weight decay of 0, so we applied this universally to limit the grid complexity.

For the token classification tasks, we selected the best-performing hyperparameters based on the seqeval (2018) F1 micro score on the validation set for Bokmål after fine-tuning an mBERT model. For sequence classification, we used the F1 macro score.

	NER	POS	Sentiment	Political
Learning Rate	$2e-5$	$3e-5$	$3e-5$	$2e-5$
Number of Epochs	3	3	3	3

Table 3: Optimal fine-tuning hyperparameters for the mBERT model using the validation datasets.

We then used the optimal fine-tuning parameters from the mBERT model for the validation dataset on our model and on the NorBERT model. Last, we compared all the models based on their results in relation to the test dataset.

Version B of our model—the version with the extended training-sequence length—performed slightly better on all four tasks than did version A. To simplify the results presented here, we therefore report only the results from version B, which we are naming *NB-BERT*.

As can be seen in the Table 4, the NB-BERT model performed significantly better than did the mBERT model for both Bokmål and Nynorsk, and on both token and sequence classification. The improvement was the smallest for the POS dataset, with an improvement from 98.3 to 98.8 for Bokmål and from 98.0 to 98.8 for Nynorsk. However, POS datasets such as this always con-

tain some ambiguity, and it is hard to tell how much more improvement is possible there. In addition, the NER task improved from 83.8 to 91.2 for Bokmål and from 85.6 to 88.9 for Nynorsk. The sequence classification improved from 69.7 to 86.4 in terms of sentiment classification and from 78.4 to 81.8 for political classification. We also tested the release 1.1 of the NorBERT model that is uploaded to Hugging Face (Language Technology Group at the University of Oslo, 2021a). The performance of this model lays in between that of NB-BERT and mBERT for Bokmål and Nynorsk, but it generally performs worse on all non-Norwegian tasks.

As shown in Table 5, our model was able to outperform the English-only and multilingual BERT for both Norwegian Bokmål and Nynorsk, as well as for Swedish and Danish, which are languages with a shared tradition with Norwegian. For English, our results are also marginally better than those obtained using the English-only BERT model. For Spanish and Finnish, for which there is no close relationship with Norwegian nor documented occurrences of text in such languages in our corpus, the mBERT model outperformed both the English-only BERT and our model, suggesting that our model is deteriorating for the languages not included in the corpus.

6 Discussion

The majority of the training corpora used today for training transformer models are built using mainly open web sources. A major motivation for this project was to investigate whether the digital collections at the NLN could be used to create a suitable corpus to train state-of-the-art transformer language models. The texts available through the library are heterogeneous in nature, including cartoons, novels, news articles, poetry, and government documents published over time and in dif-

	NER		POS		Sentiment	Political
	Bokmål	Nynorsk	Bokmål	Nynorsk	Bokmål & Nynorsk	Bokmål
mBERT	83.8	85.6	98.3	98.0	69.7	78.4
NorBERT	89.9	86.1	98.5	98.4	81.7	78.2
NB-BERT (ours)	91.2	88.9	98.8	98.8	86.4	81.8

Table 4: Evaluation results from the test dataset (version B of the model; F1 micro in token classifications and F1 macro in sequence classifications; best scores in bold).

	Bokmål	Nynorsk	English	Swedish	Danish	Spanish	Finnish
English BERT	75.1	77.8	91.3	82.5	73.9	81.8	82.9
mBERT	83.8	85.6	90.8	85.3	83.4	87.6	88.7
NorBERT	89.9	86.1	87.8	83.4	80.7	79.3	81.5
NB-BERT (ours)	91.2	88.9	91.3	85.9	85.1	85.8	85.8

Table 5: Evaluation results (F1 micro) of different monolingual NER datasets using the English-only BERT, mBERT, NorBERT, and our model (best scores in bold).

ferent contexts. As our results suggest, this seems to be a strength rather than a weakness, in that it enables us to build high-performance transformer models for small languages, such as Norwegian. Consequently, our Norwegian corpus is not only richer in diversity but also significantly larger in size than is any other Norwegian corpus, and it even rivals the size of previous work on a major language such as English. The Norwegian part of the mBERT model consists of around 1GB of text (Wu and Dredze, 2020), while the English-only BERT model was trained on 16GB of text (Devlin et al., 2019) mainly based on English Wikipedia and Open Book Corpus. When Facebook developed the first version of its RoBERTa, it added Common Crawl data and Open WebText to the BERT corpus and ended up with 160GB of text (Liu et al., 2019). Our clean corpus of Norwegian-only text is 109GB in size.

For the target languages Norwegian Bokmål and Norwegian Nynorsk, the model performs significantly better than does the mBERT model on both token classifications (POS and NER) as well as on the two sequence classification tasks. In the Bokmål NER task, the level of improvement was +7.4 F1 points. Because none of the datasets have been benchmarked against human performance, it is hard to measure how close this is to the theoretical maximum.

The results show that our corpus is a valid training source, and this is by no means surprising. All research points to the possibility of improving transformer models’ performance by training them

on larger text corpora. However, the novelty of our results lies in that we were able to increase the performance on our domain-specific tasks while maintaining a lot of the multilingual properties of the mBERT model. This was unexpected because English only comprised around 4% of the training set. Still, we were able to improve the English capabilities of the model up to the level of the monolingual English model. Part of the reason for this might be that we applied some training techniques that were not available when the English-only model was trained and released, most notably the use of larger batch sizes and the LAMB optimizer.

We were also able to significantly improve the scores for Swedish and Danish, though it is hard to pinpoint how much of this was caused by the close linguistic similarities between the languages and how much by the fact that they were represented in the corpus to some degree.

It should not be surprising that the capabilities of the model in relation to languages that were not included in the training corpus (i.e., Spanish and Finnish) did deteriorate. However, the drop in performance was not radical, and the results above indicate that we might have been able to prevent this by adding just a small portion of these languages to the large corpus.

Overall, our results suggest that collections such as the digital collection at the NLN, even if they contain occasional OCR-errors, may contribute significantly toward the creation of well-performing language models by providing large

training corpora. As discussed earlier, there are OCR errors in the included materials. An exhaustive removal of all OCR artifacts would either have required us to do a major reduction of the size of the corpus, or to invest an unmanageable amount of manual work. We have not seen any indication that the OCR errors negatively impacted the performance. We might speculate that the model has learned to distinguish OCR errors from ordinary text, indicating that quantity is more important than quality when building such corpora. All in all, size matters.

7 Conclusion and Future Work

In this work, we have investigated the feasibility of building a large Norwegian-only corpus for the training of well-performing transformer-based language models. We relied on the collections of the NLN, and our model outperformed the existing multilingual alternatives. In the process, while the corpus produced might lack the cleanness of other textual resources, we proved that using somewhat noisy but available sources is an effective way to grow the ecosystem of resources for languages with fewer resources and for which enough open text in a digital format simply does not exist. As part of an effort to democratize the use of technology and digital resources at the NLN, we are releasing our trained BERT-based model (National Library of Norway AI Lab, 2021a) and will be releasing other models based on the same corpus in the future. Moreover, we are also releasing the set of tools and code we used so that others seeking similar results can easily reuse them (National Library of Norway AI Lab, 2021b).

Although our work may indicate that OCR errors in corpora have little to no impact on the quality of the resulting transformer model, this has not been explicitly proven in the current study. More systematic studies are needed to investigate the real effect of OCR noise and artifacts.

Another important aspect is that, to benefit from the pre-trained mBERT weights, we used a 119,547-token multilingual vocabulary, of which only a small fraction pertained to Norwegian. A natural follow up would be to investigate the performance gains of using only a tailored Norwegian vocabulary.

The decision to use a BERT-based architecture as our target was guided by its simplicity to train and benchmark. However, newer and better-

performing models have been released since the original BERT work a few years ago. The current corpus could be used for training such models as well studying the differences between architectural styles and training objectives. While it is already large in size, there is still potential to grow our 109GB corpus to the limits of the extant Norwegian holdings at the NLN, which presents itself as an opportunity to release even larger models.

Funding

This research was supported by Cloud TPUs from Google’s TPU Research Cloud (TRC).

Acknowledgment

We would like to thank KBLab at the National Library of Sweden (Kungliga biblioteket) for its pioneering work on BERT in memory institutions and for the valuable and inspiring discussions. We also appreciate the feedback from and discussions with Andre Kaasen of the Language Bank at the National Library of Norway.

References

- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- BotXO Ltd. 2021. https://github.com/botxo/nordic_bert Pre-trained nordic models for bert. [Online; accessed 5-February-2021].
- Common Crawl Foundation. 2021. <https://commoncrawl.org/> Common crawl. [Online; accessed 5-February-2021].
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. <https://doi.org/10.18653/v1/N19-1423> BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jan Terje Faarlund, Svein Lie, and Kjell Ivar Vannebo. 1997. *Norsk referansegrammatikk*. Universitetsforlaget.
- Gothenburg University Språkbanken. 2012. Swedish ner corpus, 2012.

- Rasmus Hvingelby, Amalie Brogaard Pauli, Maria Barrett, Christina Rosted, Lasse Malm Lidegaard, and Anders Søgaard. 2020. <https://www.aclweb.org/anthology/2020.lrec-1.565> DaNE: A named entity resource for Danish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4597–4604, Marseille, France. European Language Resources Association.
- Janne Bondi Johannessen, Kristin Hagen, Lynum André, and Anders. Nøklestad. 2012. Obt+stat. a combined rule-based and statistical tagger. In *Exploring Newspaper Language. Corpus Compilation and Research Based on the Norwegian Newspaper Corpus*, pages 51–65.
- Fredrik Jørgensen, Tobias Aasmoe, Anne-Stine Ruud Husevåg, Lilja Øvrelid, and Erik Velldal. 2020. NorNE: Annotating Named Entities for Norwegian. In *Proceedings of the 12th Edition of the Language Resources and Evaluation Conference*, Marseille, France.
- Andrei Kutuzov, Murhaf Fares, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 58th Conference on Simulation and Modelling*, pages 271–276. Linköping University Electronic Press.
- Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-scale contextualised language modelling for norwegian. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa 2021)*.
- Language Technology Group at the University of Oslo. 2021a. <https://huggingface.co/lrgoslo/norbert> NorBERT: Hugging face page. Release 1.1. February 13, 2021.
- Language Technology Group at the University of Oslo. 2021b. https://github.com/lrgoslo/norec_sentence/ Norec sentence.
- Emanuele Lapponi, Martin G. Søyland, Erik Velldal, and Stephan Oepen. 2018. <https://doi.org/10.1007/s10579-018-9411-5> The Talk of Norway: a richly annotated corpus of the Norwegian parliament, 1998–2016. *Language Resources and Evaluation*, pages 1–21.
- Library of Congress. 2016. <https://www.loc.gov/standards/alto/> Alto. [Online; accessed 5-February-2021].
- Library of Congress. 2020. <https://www.loc.gov/standards/mets/> Mets. [Online; accessed 5-February-2021].
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. <http://arxiv.org/abs/2007.01658> Playing with words at the national library of sweden – making a swedish bert.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations*.
- Hiroki Nakayama. 2018. <https://github.com/chakki-works/seqeval> seqeval: A python framework for sequence labeling e evaluation. [Online; accessed 5-February-2021].
- National Library of Norway AI Lab. 2021a. <https://huggingface.co/NbAiLab/nb-bert-base> Nbailab/nb-bert-base · hugging face. [Online; accessed 5-February-2021].
- National Library of Norway AI Lab. 2021b. <https://github.com/NbAiLab/notram> Nbailab/notram: Norwegian transformer model. [Online; accessed 5-February-2021].
- Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2020. <https://www.aclweb.org/anthology/2020.lrec-1.618> A fine-grained sentiment dataset for Norwegian. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5025–5033, Marseille, France. European Language Resources Association.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.
- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. <https://doi.org/10.18653/v1/P19-1493> How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.
- Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. 2019. A finnish news corpus for named entity recognition. *Language Resources and Evaluation*, pages 1–26.
- Per Erik Solberg, Arne Skjærholt, Lilja Øvrelid, Kristin Hagen, and Janne Bondi Johannessen. 2014. The norwegian dependency treebank.
- Språkbanken. 2019. <https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-4/> Norsk aviskorpus. [Online; accessed 5-February-2021].

Språkbanken. 2021. <https://www.nb.no/sprakbanken/en/sprakbanken/> Språkbanken — the norwegian language bank. [Online; accessed 5-February-2021].

TensorFlow Hub. 2021. https://tfhub.dev/tensorflow/bert_multi_cased_L-12_H-768_A-12/3_mBERT_TFHub. [Online; accessed 5-February-2021].

Erik F. Tjong Kim Sang. 2002. <https://www.aclweb.org/anthology/W02-2024> Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. <https://www.aclweb.org/anthology/W03-0419> Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Wikipedia Contributors. 2021a. https://en.wikipedia.org/wiki/Wikipedia:Database_download Wikipedia:database download — Wikipedia, the free encyclopedia. [Online; accessed 5-February-2021].

Wikipedia Contributors. 2021b. https://meta.wikimedia.org/wiki/List_of_Wikipedias List of wikipedia — Wikipedia, the free encyclopedia. [Online; accessed 5-February-2021].

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual bert? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2019. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 19–27.

Large-Scale Contextualised Language Modelling for Norwegian

Andrey Kutuzov, Jeremy Barnes, Erik Veldal,
Lilja Øvrelid and Stephan Oepen

University of Oslo
Department of Informatics
Language Technology Group

{andreku|jeremycb|erikve|liljao|oe}@ifi.uio.no

Abstract

We present the ongoing NorLM initiative to support the creation and use of very large contextualised language models for Norwegian (and in principle other Nordic languages), including a ready-to-use software environment, as well as an experience report for data preparation and training. This paper introduces the first large-scale monolingual language models for Norwegian, based on both the ELMo and BERT frameworks. In addition to detailing the training process, we present contrastive benchmark results on a suite of NLP tasks for Norwegian.

For additional background and access to the data, models, and software, please see:

<http://norlm.nlpl.eu>

1 Introduction

In this work, we present *NorLM*, an ongoing community initiative and emerging collection of large-scale contextualised language models for Norwegian. We here introduce the NorELMo and NorBERT models, that have been trained on around two billion tokens of running Norwegian text. We describe the training procedure and compare these models with the multilingual mBERT model (Devlin et al., 2019), as well as an additional Norwegian BERT model developed contemporaneously, with some interesting differences in training data and setup. We report results over a number of Norwegian benchmark datasets, addressing a broad range of diverse NLP tasks: part-of-speech tagging, negation resolution, sentence-level and fine-grained sentiment analysis and named entity recognition (NER).

All the models are publicly available for download from the Nordic Language Processing Lab-

oratory (NLPL) Vectors Repository¹ with a CC BY 4.0 license. They are also accessible locally, together with the training and supporting software, on the two national superclusters Puhti and Saga, in Finland and Norway, respectively, which are available to university NLP research groups in Northern Europe through the Nordic Language Processing Laboratory (NLPL).² The NorBERT model is in addition served via the Huggingface Transformers model hub.³

NorLM is a joint effort of the projects EOSC-Nordic (European Open Science Cloud) and SANT (Sentiment Analysis for Norwegian), coordinated by the Language Technology Group (LTG) at the University of Oslo. The goal of this work is to provide these models and supporting tools for researchers and developers in Natural Language Processing (NLP) for the Norwegian language. We do so in the hope of facilitating scientific experimentation with and practical applications of state-of-the-art NLP architectures, as well as to enable others to develop their own large-scale models, for example for domain- or application-specific tasks, language variants, or even other languages than Norwegian. Under the auspices of the NLPL use case in EOSC-Nordic, we are also coordinating with colleagues in Denmark, Finland, and Sweden on a collection of large contextualised language models for the Nordic languages, including language variants or related groups of languages, as linguistically or technologically appropriate.

2 Background

Bokmål and Nynorsk There are two official standards for written Norwegian; *Bokmål*, the main variety, and *Nynorsk*, used by 10–15% of

¹<http://vectors.nlpl.eu/repository>

²<http://www.nlpl.eu>

³<https://huggingface.co/ltgoslo/norbert>

the Norwegian population. Norwegian language legislation specifies that minimally 25% of the written public service information should be in Nynorsk. While the two varieties are closely related, there can also be relatively large differences lexically (though often with a large degree of overlap on the character-level still). Several previous studies have indicated that joint modeling of Bokmål and Nynorsk works well for many NLP tasks, like tagging and parsing (Velldal et al., 2017) and NER (Jørgensen et al., 2020). The contextualised language models presented in this paper are therefore trained jointly on both varieties, but with the minority variant Nynorsk represented by comparatively less data than Bokmål (reflecting the natural usage).

Datasets For all our models presented below, we used the following training corpora:

1. Norsk Aviskorpus (NAK), a collection of Norwegian news texts⁴ (both Bokmål and Nynorsk) from 1998 to 2019; 1.7 billion words;
2. Bokmål Wikipedia dump from September 2020; 160 million words;
3. Nynorsk Wikipedia dump from September 2020; 40 million words.

The corpora contain ordered sentences (which is important for BERT-like models, because one of their training tasks is next sentence prediction). In total, our training corpus comprises about two billion (1,907,072,909) word tokens in 203 million (202,802,665) sentences.

We conducted the following pre-processing steps:

1. Wikipedia texts were extracted from the dumps using the `segment_wiki` script from the Gensim project (Řehůřek and Sojka, 2010).
2. For the news texts from Norwegian Aviskorpus, we performed de-tokenization and conversion to UTF-8 encoding, where required.
3. The resulting corpus was sentence-segmented using Stanza (Qi et al., 2020). We left blank lines between documents (and

sections in the case of Wikipedia) so that the ‘next sentence prediction’ task of BERT does not span between documents.

3 Prerequisites: software and computing

Developing very large contextualised language models is no small challenge, both in terms of engineering sophistication and computing demands. Training ELMO- and in particular BERT-like models presupposes access to specialised hardware – graphical processing units (GPUs) – over extended periods of time. Compared to the original work at Google or to our sister initiative at the National Library of Norway (see below), our two billion tokens in Norwegian training data can be characterised as moderate in size.

Nevertheless, training a single NorBERT model requires close to one full year of GPU utilisation, which through parallelization over multiple compute nodes, each featuring four GPUs, could be completed in about three weeks of wall clock time. At this scale, premium software efficiency and effective parallelization are prerequisites, not only to allow repeated incremental training and evaluation cycles to complete in practical intervals, but equally so for cost-efficient utilisation of scarce, shared computing resources and, ultimately, a shred of environmental sustainability.

To prepare the NorLM software environment, we have teamed up with support staff at the Norwegian national e-infrastructure provider, Uninett Sigma2, and developed a fully automated and modularised installation procedure using the Easy-Build framework (<https://easybuild.io>). All necessary tools are compiled from source with the right set of hardware-specific optimizations and platform-specific optimised libraries for basic linear algebra (‘math kernels’) and communication across multiple compute nodes.

This approach to software provisioning makes it possible to (largely) automatically create fully parallel training and experimentation environments on multiple computing infrastructures – in our work to date two national HPC superclusters, in Norway and Finland, but in principle just as much any suitable local GPU cluster. In our view, making available both a ready-to-run software environment on Nordic national e-infrastructures, where university research groups typically can gain no-cost access, coupled with the recipe for recreating the environment on other HPC systems, may

⁴<https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-4/>

contribute to ‘democratising’ large-scale NLP research; if nothing else, it eliminates dependency on commercial cloud computing services.

4 Related work

Large-scale deep learning language models (LM) are important components of current NLP systems. They are often based on BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) and other contextualised architectures. A number of language-specific initiatives have in recent years released monolingual versions of these models for a number of languages (Fares et al., 2017; Kutuzov and Kuzmenko, 2017; Virtanen et al., 2019; de Vries et al., 2019; Ulčar and Robnik-Šikonja, 2020; Koutsikakis et al., 2020; Nguyen and Nguyen, 2020; Farahani et al., 2020; Malmsten et al., 2020). For our purposes, the most important such previous training effort is that of Virtanen et al. (2019) on creating a BERT model for Finnish – FinBERT⁵ – as our training setup for creating NorBERT builds heavily on this; see Section 6 for more details.

Many low-resource languages do not have dedicated monolingual large-scale language models, and instead resort to using a multilingual model, such as Google’s multilingual BERT model – mBERT – which was trained on data that also included Norwegian. Up until the release of the models described in the current paper, mBERT was the only BERT-instance that could be used for Norwegian.⁶

Another widely used architecture for contextualised LMs is Embeddings From Language Models or ELMo (Peters et al., 2018). The *ElmoForManyLangs* initiative (Che et al., 2018) trained and released monolingual ELMo models for a wide range of different languages, including Norwegian (with separate models for Bokmål and Nynorsk). However, these models were trained on very modestly sized corpora of 20 million words for each language (randomly sampled from Wikipedia dumps and Common Crawl data).

In a parallel effort to that of the current paper, the AI Lab of the National Library of Norway, through their Norwegian Transformer Model (No-

TraM) project, has released a Norwegian BERT (Base, cased) model dubbed NB-BERT (Kummer-vold et al., 2021).⁷ The model is trained on the Colossal Norwegian Corpus, reported to comprise close to 18,5 billion words (109.1 GB of text).

In raw numbers, this is about ten times more than the corpus we use for training the NorLM models. However, the vast majority of this is from OCR’ed historical sources, which is bound to introduce at least some noise. In Section 7 below, we demonstrate that in some NLP tasks, a language model trained on less (but arguably cleaner) data can outperform a model trained on larger but noisy corpora.

5 NorELMo

NorELMo is a set of bidirectional recurrent ELMo language models trained from scratch on the Norwegian corpus described in Section 1. They can be used as a source of contextualised token representations for various Norwegian natural language processing tasks. As we show below, in many cases, they present a viable alternative to Transformer-based models like BERT. Their performance is often only marginally lower, while the compute time required to adapt the model to the task at hand can be an order of magnitude less on identical hardware.

Currently we present two models, with more following in the future:

1. **NorELMo₃₀**: 30,000 most frequent words in the vocabulary
2. **NorELMo₁₀₀**: 100,000 most frequent words in the vocabulary

Note that independent of the vocabulary size, both NorELMo₃₀ and NorELMo₁₀₀ can process arbitrary word tokens, due to the ELMo architecture (where the first CNN layer converts input strings to non-contextual word embeddings). Thus, the size of the vocabulary controls only the number of words used as targets for the language modelling task in the course of training. Supposedly, the model with a larger vocabulary is more effective in treating less frequent words at the cost of being less effective with more frequent words.

Each model was trained for 3 epochs with batch size 192. We employed a version of the original

⁵<https://github.com/TurkuNLP/FinBERT>

⁶A BERT model trained on Norwegian data was published at https://github.com/botxo/nordic_bert in the beginning of 2020. However, the vocabulary of this model seems to be broken, and to the best of our knowledge nobody has achieved any meaningful results with it.

⁷<https://github.com/NBAiLab/notram>

ELMo training code from Peters et al. (2018) updated to work better with the recent TensorFlow versions. All the hyperparameters were left at their default values, except the LSTM dimensionality reduced to 2,048 from the default 4,096 (in our experience, this rarely influences performance). Training of each model took about 100 hours on four NVIDIA P100 GPUs.

These are the first ELMo models for Norwegian trained on a large corpus. As has already been mentioned, the Norwegian ELMo models from the *ElmoForManyLangs* project (Che et al., 2018) were trained on very small corpora samples and seriously under-perform on semantic-related NLP tasks, although they can yield impressive results on POS tagging and syntactic parsing (Zeman et al., 2018). In addition, they were trained with custom code modifications and can be used only with the custom *ElmoForManyLangs* library. On the other hand, our NorELMo models are fully compatible both with the original ELMo implementation by Peters et al. (2018) and with the more modern *simple_elmo* Python library provided by us.⁸

The vocabularies are published together with the models. For different tasks, different models can be better, as we show below. The published packages contain both TensorFlow checkpoints (for possible fine-tuning, if need be) and model files in the standard Hierarchical Data Format (HDF5) for easier inference usage. In addition, we have setup ELMoViz, a demo web service to explore Norwegian ELMo models.⁹

6 NorBERT

Our NorBERT model is trained from scratch for Norwegian, and can be used in exactly the same way as any other BERT-like model. The NorBERT training setup heavily builds on prior work on FinBERT conducted at the University of Turku (Virtanen et al., 2019).

NorBERT features a custom WordPiece vocabulary which is case-sensitive and includes accented characters. It has much better coverage of Norwegian words than the mBERT model or NB-BERT (which uses the same vocabulary as mBERT). This is clearly seen on the example of the tokenization performed by both for the Norwe-

gian sentence ‘*Denne gjengen håper at de sammen skal bidra til å gi kvinnefotballen i Kristiansand et lenge etterlengtet løft*’

- **mBERT/NB-BERT:** ‘Denne g ##jeng ##en h ##å ##per at de sammen skal bid ##ra til å gi k ##vinne ##fo ##t ##ball ##en i Kristiansand et lenge etter ##len ##gte ##t l ##ø ##ft’
- **NorBERT:** ‘Denne gjengen håper at de sammen skal bidra til å gi kvinne ##fotball ##en i Kristiansand et lenge etterl ##engt ##et løft’

NorBERT tokenization splits the sentence into pieces which much better reflect the real Norwegian words and morphemes (cf. ‘*k vinne fo t ball en*’ versus ‘*kvinne fotball en*’). We believe this to be extremely important for more linguistically-oriented studies, where it is critical to deal with words, not with arbitrarily fragmented pieces (even if they are well-performing in practical tasks).

The vocabulary for the model is of size 30,000. It is much less than the 120,000 of mBERT, but it is compensated by these entities being almost exclusively Norwegian. The vocabulary was generated from raw text, without, e.g., separating punctuation from word tokens. This means one can feed raw text into NorBERT.

For the vocabulary generation, we used the SentencePiece algorithm (Kudo, 2018) and Tokenizers library.¹⁰ The resulting Tokenizers model was converted to the standard BERT WordPiece format. The final vocabulary contains several thousand unused wordpiece slots which can be filled in with task-specific lexical entries for further fine-tuning by future NorBERT users.

6.1 Training technicalities

NorBERT corresponds in its configuration to the Google’s Bert-Base Cased for English, with 12 layers and hidden size 768 (Devlin et al., 2019). We used the standard masked language modeling and next sentence prediction losses with the LAMB optimizer (You et al., 2020). The model was trained on the Norwegian academic HPC system called Saga. Most of the time the training process was distributed across 4 compute nodes and 16 NVIDIA P100 GPUs. Overall, it took approximately 3 weeks (more than 500 hours).

⁸<https://pypi.org/project/simple-elmo/>

⁹<http://vectors.nlpl.eu/explore/embeddings/en/contextual/>

¹⁰<https://github.com/huggingface/tokenizers>

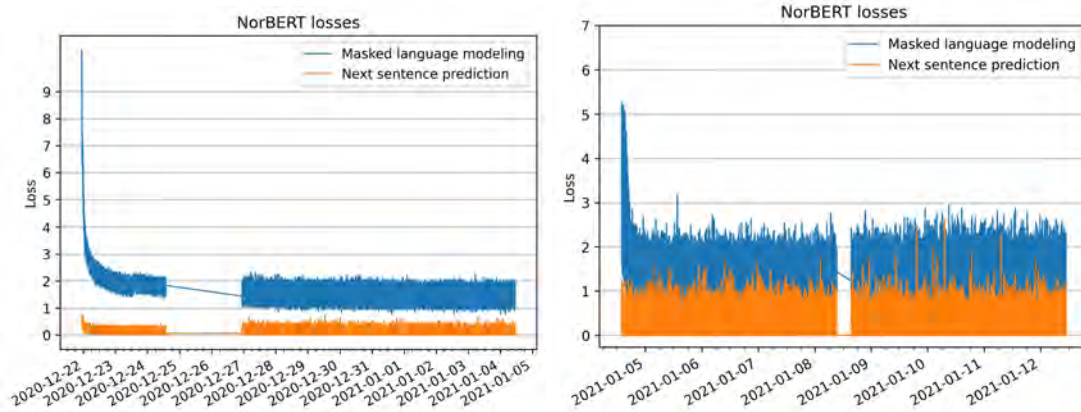


Figure 1: NorBERT loss plots at the Phase 1 (left) and Phase 2 (right).

Similar to Virtanen et al. (2019), we employed the BERT implementation by NVIDIA¹¹, which allows fast multi-node and multi-GPU training.

We made minor changes to this code, mostly to adapt it to the newer TensorFlow versions. All these patches and the utilities we used at the pre-processing, training and evaluation stages are published in our GitHub repository.¹² Instructions to reproduce the training setup with the EasyBuild software build and installation framework are also available.¹³

6.2 Training workflow

Phase 1 (training with maximum sequence length of 128) was done with batch size 48 and global batch size $48 \cdot 16 = 768$. Since one global batch contains 768 sentences, approximately 265,000 training steps constitute 1 epoch (one pass over the whole corpus). We have done 3 epochs: 795,000 training steps.

Phase 2 (training with maximum sequence length of 512) was done with batch size 8 and global batch size $8 \cdot 16 = 128$. We aimed at mimicking the original BERT in that at Phase 2 the model should see about 1/9 of the number of sentences seen during Phase 1. Thus, we needed about 68 million sentences, which at the global batch size of 128 boils down to 531,000 training steps more.

The loss plots are shown in Figure 1 (the training was on pause on December 25 and 26, since we were solving problems with mixed precision

¹¹<https://github.com/NVIDIA/DeepLearningExamples/tree/master/TensorFlow/LanguageModeling/BERT>, version 20.06.08

¹²<https://github.com/lgtoslo/NorBERT>

¹³<http://wiki.nlpl.eu/index.php/Eosc/pretraining/nvidia>

Task	Train	Dev	Test
POS Bokmål	15,696	2,409	1,939
POS Nynorsk	14,174	1,890	1,511
NER Bokmål	15,696	2,409	1,939
NER Nynorsk	14,174	1,890	1,511
Sentence-level SA	2,675	516	417
Fine-grained SA	8,543	1,531	1,272
Negation	8,543	1,531	1,272

Table 1: Number of sentences in the training, development, and test splits in the datasets used for the evaluation tasks.

training). Full logs are available at the GitHub repository.

7 Evaluation

This section presents benchmark results across a range of different tasks. We compare NorELMO and NorBERT to both mBERT and to the recently released NB-BERT model described in Section 4. Where applicable, we show separate evaluation results for Bokmål and Nynorsk. Below we first provide an overview of the different tasks and the corresponding classifiers that we train, before turning to discuss the results.

7.1 Task descriptions

We start by briefly describing each task and associated dataset, in addition to the architectures we use. The sentence counts for the different datasets and their train, dev. and test splits are provided in Table 1.

Part-of-speech tagging The Norwegian Dependency Treebank (NDT) (Solberg et al., 2014) in-

cludes annotation of POS tags for both Bokmål and Nynorsk. NDT has also been converted to the Universal Dependencies format (Øvrelid and Hohle, 2016; Velldal et al., 2017) and this is the version we are using here (for UD 2.7) for predicting UPOS tags.

We use a typical sequence labelling approach with the BERT models, adding a linear layer after the final token representations and taking the softmax to get token predictions. We fine-tune all parameters for 20 epochs, using a learning rate of $2e-5$, a training batch size of 8, max length of 256, and keep the best model on the development set. ELMo models were not fine-tuned, following the recommendations from Peters et al. (2019). Instead we trained a simple neural classifier (a feed forward network with one hidden layer of size 128, ReLU non-linear activation function and dropout), using ELMo token embeddings as features. The random seed has been kept fixed all the time. Models are evaluated on accuracy.

Named entity recognition The NorNE¹⁴ dataset annotates the UD-version of NDT with a rich set of entity types (Jørgensen et al., 2020). The evaluation metrics here is ‘strict’ micro F_1 , requiring both the correct entity type and exact match of boundary surface string. We predict 8 entity types: Person (PER), Organisation (ORG), Location (LOC), Geo-political entity, with a locative sense (GPE-LOC), Geo-political entity, with an organisation sense (GPE-ORG), Product (PROD), Event (EVT), Nominals derived from names (DRV). The evaluation is done using the code for the SemEval’13 Task 9¹⁵.

We cast the named entity recognition problem as a sequence labelling task, using a BIO label encoding. For the BERT-based models, we solve it by fine-tuning the pre-trained model on the NorNE dataset for 20 epochs with early stopping and batch size 32. The resulting model is applied to the test set.

For ELMo models, we infer contextualised token embeddings (averaged representations across all 3 layers) for all words. Then, these token embeddings are fed to a neural classifier with dropout, identical to the one we used for POS tagging earlier. This classifier is also trained for 20 epochs with early stopping and batch size 32.

¹⁴<https://github.com/ltgoslo/norne>

¹⁵<https://github.com/davidsbatista/NER-Evaluation>

Fine-grained sentiment analysis NoReC_{fine} is a dataset¹⁶ comprising a subset of the Norwegian Review Corpus (NoReC; Velldal et al., 2018) annotated for sentiment holders, targets, expressions, and polarity, as well as the relationships between them (Øvrelid et al., 2020). We here cast the problem as a graph prediction task and train a graph parser (Dozat and Manning, 2018; Kurtz et al., 2020) to predict sentiment graphs. The parser creates token-level representations which is the concatenation of a word embedding, POS tag embedding, lemma embedding, and character embedding created by a character-based LSTM. We further augment these representations with contextualised embeddings from each model. Models are trained for 100 epochs, keeping the best model on development F_1 . For span extraction (holders, targets, expressions), we evaluate token-level F_1 , and the common Targeted F_1 metric, which requires correctly extracting a target (strict) and its polarity. We also evaluate Labelled and Unlabelled F_1 , which correspond to Labelled and Unlabelled Attachment in dependency parsing. Finally, we evaluate on Sentiment Graph F_1 (SF_1) and Non-polar Sentiment Graph F_1 (NSF_1). SF_1 requires predicting all elements (holder, target, expression, polarity) and their relationships (NSF_1 removes the polarity). A true positive is defined as an exact match at graph-level, weighting the overlap in predicted and gold spans for each element, averaged across all three spans. For precision we weight the number of correctly predicted tokens divided by the total number of predicted tokens (for recall, we divide instead by the number of gold tokens). We allow for empty holders and targets.

Sentence-level binary sentiment classification

We further evaluate on the task of sentence-level binary (positive or negative) polarity classification, using labels that we derive from NoReC_{fine} described above. We create the dataset for this by aggregating the fine-grained annotations to the sentence-level, removing sentences with mixed or no sentiment. The resulting dataset, NoReC_{sentence}, is made publicly available.¹⁷ For the BERT models, we use the [CLS] embedding of the last layer as a representation for the sentence and pass this to a softmax layer for classification. We fine-tune the models in the same way as for

¹⁶https://github.com/ltgoslo/norec_fine

¹⁷https://github.com/ltgoslo/norec_sentence

Model	POS		Time
	BM	NN	
Stanza (Qi et al., 2020)	98.3	97.9	–
NorELMo ₃₀	98.1	97.4	8
NorELMo ₁₀₀	98.0	97.4	8
mBERT	98.0	97.9	245
NB-BERT	98.7	98.3	244
NorBERT	98.5	98.0	238

Table 2: Evaluation scores of the NorLM models on the POS tagging of Bokmål (BM) and Nynorsk (NN) test sets in comparison with other large pre-trained models for Norwegian. Running times in minutes are given for Bokmål.

the POS tagging task, training the models for 20 epochs and keeping the model that performs best on the development data. For ELMo models, we used a BiLSTM with global max pooling, taking ELMo token embeddings from the top layer as an input. The evaluation metric is macro F_1 .

Negation detection Finally, the NoReC_{fine} dataset has recently been annotated with negation cues and their corresponding in-sentence scopes (Mæhlum et al., 2021). The resulting dataset is dubbed NoReC_{neg}.¹⁸ We use the same graph-based modeling approach as described for fine-grained sentiment above. We evaluate on the same metrics as in the *SEM 2012 shared task (Morante and Blanco, 2012): cue-level F_1 (CUE), scope token F_1 over individual tokens (ST), and the combined full negation F_1 (FN).

7.2 Results

We present the results for the various benchmarking tasks below.

POS tagging As can be seen from Table 2, NorBERT outperforms mBERT on both tasks: on POS tagging for Bokmål by 5 percentage points and 1 percentage point for Nynorsk. NorBERT is almost on par with NB-BERT on POS tagging. NorELMo models are outperformed by NB-BERT and NorBERT, but are on par with mBERT in POS tagging. Note that their adaptation to the tasks (extracting token embeddings and learning a classifier) takes 30x less time than with the BERT models.

¹⁸https://github.com/lsgoslo/norec_neg

Model	Bokmål	Nynorsk	Time
NorELMo ₃₀	79.9	75.6	2
NorELMo ₁₀₀	81.3	75.1	2
mBERT	78.8	81.7	14
NB-BERT	90.2	88.6	11
NorBERT	85.5	82.8	9

Table 3: NER evaluation scores (micro F_1) of the NorLM models on the NorNE test set in comparison with other large pre-trained models for Norwegian. Running time is given in minutes for the Bokmål part (on 1 NVIDIA P100 GPU).

See Figure 2 for the examples of training dynamics of the Nynorsk model.

Named entity recognition Table 3 shows the performance on the NER task. NB-BERT is the best on both Bokmål and Nynorsk, closely followed by NorBERT. Unsurprisingly, mBERT falls behind all the models trained for Norwegian, when evaluated on Bokmål data. With Nynorsk, it manages to outperform NorELMo. Bokmål is presumably dominant in the training corpora of both. However, in the course of fine-tuning, mBERT seems to be able to adapt to the specifics of Nynorsk. Since our ELMo setup did not include the fine-tuning step, the NorELMo models’ adaptation abilities were limited by what can be learned from contextualised token embeddings produced by a frozen model. Still, when used on the data more similar to the training corpus (Bokmål), ELMo achieves competitive results even without any fine-tuning.

In terms of computational efficiency, the adaptation of ELMo models to this task requires 6x less time than mBERT or NB-BERT and 4x less time than NorBERT. Note also that the NorBERT model takes less time to fine-tune than the NB-BERT model (although the number of epochs was exactly the same), because of a smaller vocabulary, and thus less parameters in the model. Again, in this case an NLP practitioner has a rich spectrum of tools to choose from, depending on whether speed or performance on the downstream task is prioritised.

Fine-grained sentiment analysis Table 4 shows that NorBERT outperforms mBERT on all metrics and NB-BERT on all but SF₁, although the differences between NorBERT and NB-BERT are gen-

Model	Spans			Targeted	Parsing Graph		Sent. Graph		Time
	Holder F ₁	Target F ₁	Exp. F ₁	F ₁	UF ₁	LF ₁	NSF ₁	SF ₁	
Extraction [1]	42.4	31.3	31.3	–	–	–	–	–	–
NorELMo ₃₀	55.1	55.3	57.2	37.9	49.0	41.2	40.9	34.5	446
NorELMo ₁₀₀	58.8	55.8	56.8	37.1	49.7	41.2	41.5	34.2	434
mBERT	57.1	55.2	56.3	34.8	48.7	38.3	40.5	31.7	444
NB-BERT	61.3	56.1	57.9	36.0	49.7	41.9	40.7	34.8	404
NorBERT	63.0	56.4	58.1	36.9	50.5	42.2	41.0	34.8	438

Table 4: Average score of NorLM models on fine-grained sentiment (5 runs with set random seeds). **Bold** denotes the best result on each metric. [1] Span extraction baseline from Øvrelid et al. (2020), which uses a BiLSTM CRF with pretrained fastText embeddings.

Model	F ₁
NorELMo ₃₀	75.0
NorELMo ₁₀₀	75.0
mBERT	67.7
NB-BERT	83.9
NorBERT	77.1

Table 5: F₁ scores for the different LMs models on the binary sentiment classification test set.

Model	CUE	ST	FN	Time
NorELMo ₃₀	91.7	80.6	63.8	428
NorELMo ₁₀₀	92.2	81.3	65.5	407
mBERT	92.8	84.0	65.9	353
NB-BERT	92.4	83.1	63.5	342
NorBERT	92.1	83.6	65.5	426

Table 6: Results of our negation parser, augmenting the features with token representations from each language model. The results are averaged over 5 runs.

erally small.

On this task the NorELMo models generally outperform mBERT as well. However, unlike in the previous tasks, the running times here are similar for BERT and ELMo models, since no fine-tuning was applied (the same is true for negation detection). We furthermore compare with the previous best model (Øvrelid et al., 2020), a span extraction model which uses a single-layer Bidirectional LSTM with Conditional Random Field inference, and an embedding layer initialized with fastText vectors trained on the NoWaC corpus. All approaches using language models outperform the

previous baseline by a large margin on the span extraction tasks.¹⁹ NorBERT, in particular, achieves improvements of 20.6 percentage points on Holder F₁ (24.9 and 25.8 on Target and Exp. F₁, respectively).

Binary sentiment classification Table 5 shows that NorBERT outperforms mBERT by 9.4 percentage points on sentiment analysis. However, it seems that in binary sentiment classification the sheer amount of training data starts to show its benefits, and NB-BERT outperforms NorBERT by 6.8 points. NorELMo models outperform mBERT by 7.3 points.

Figure 2 shows the training dynamics of the models.

Negation detection From Table 6 we can see that mBERT gives the best overall results, followed by NorBERT and NorELMo₁₀₀. NB-BERT and NorELMo₃₀ perform worse than the others on Scope token F₁ (ST) and full negation F₁ (FN), while all models perform similarly at cue-level F₁ (CUE). We hypothesise that the structural similarity of negation across many of the pretraining languages gives mBERT an advantage, but it is still surprising that it outperforms NB-BERT and NorBERT.

8 Future plans

In the future, separate Bokmål and Nynorsk BERT models are planned, and we further expect to train and evaluate models with a higher number of epochs over the training corpus. While we plan to develop additional monolingual Norwegian models based on other contextualised LM architectures

¹⁹Øvrelid et al. (2020) only perform span extraction. Therefore, it is not possible to compare the other metrics.

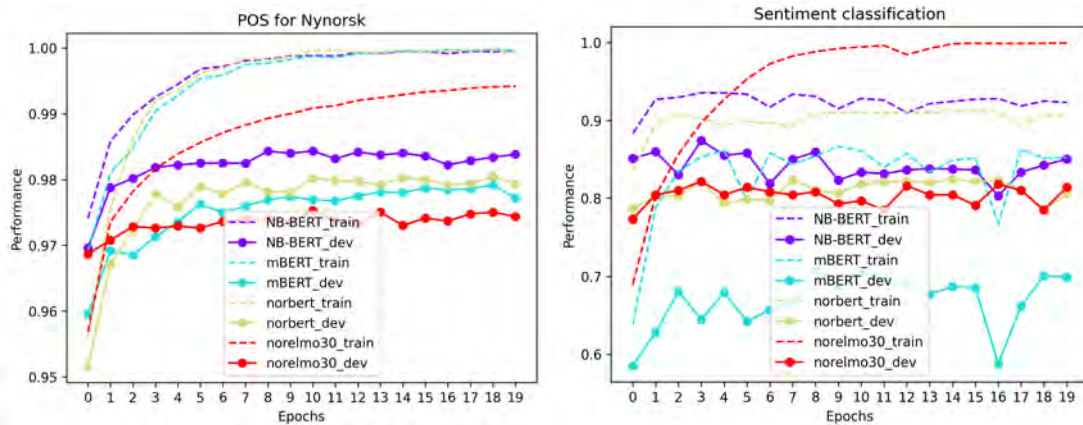


Figure 2: Per-epoch performance on training and development data for two of the tasks. Left: accuracy for POS tagging (Norwegian Nynorsk). Right: F_1 for binary sentiment classification.

beyond BERT and ELMo, we would also be interested to explore the usefulness of multilingual models restricted to Scandinavian languages. Further streamlining of the benchmarking process, in terms of both data access and computation of metrics, is something we also want to address in future work.

In addition, the ready availability of a highly optimised software stack on multiple HPC systems (published as part of NorLM) may contribute to other researchers developing very large contextualised language models for additional languages or language variants, e.g. domain- or application-specific sub-corpora. We hope that more pre-trained NLP models for Norwegian from both academy and industry will be openly released, making it possible to study the interplay between training corpora sizes, hyperparameters, pre-preprocessing decisions and performance in different tasks. At the same time, given the resource demands and sustainability issues related to training such models, we believe it will be important to coordinate efforts and we hope to collaborate closely with other players moving forward.

9 Summary

This paper has described the first outcomes of NorLM, an initiative coordinated by the Language Technology Group at the University of Oslo seeking to provide Norwegian (and Nordic) large-scale contextualised language models, while simultaneously focusing on maintaining a re-usable software environment for model development on national and Nordic HPC infrastructure. We have here described the training and testing of

NorELMo and NorBERT – the first large-scale monolingual LMs for Norwegian. We have benchmarked the models across a wide array of Norwegian NLP tasks, also comparing to the multilingual mBERT model and another large-scale LM for Norwegian developed in parallel work, NB-BERT, trained on large amounts of text from historical sources. The results show that while the monolingual models tend to yield better results, which particular model ranks first varies across tasks. This underscores the importance of building an ecosystem of diversified models, accompanied by systematic benchmarking.

Acknowledgements

The NorLM resources are being developed on the Norwegian national super-computing services operated by UNINETT Sigma2, the National Infrastructure for High Performance Computing and Data Storage in Norway, as well as on the Finnish national supercomputing facilities operated by the CSC IT Center for Science. Software provisioning was financially supported through the European EOSC-Nordic project; data preparation and evaluation were supported by the SANT project (Sentiment Analysis for Norwegian Text), funded by the Research Council of Norway (grant number 270908). We are indebted to all funding agencies involved, the University of Oslo, and the Norwegian tax payer.

References

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing:

- Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.
- Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, and Mohammad Manthouri. 2020. Parsbert: Transformer-based model for Persian language understanding. *arXiv preprint arXiv:2005.12515*.
- Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 271–276, Gothenburg, Sweden. Association for Computational Linguistics.
- Fredrik Jørgensen, Tobias Aasmoe, Anne-Stine Ruud Husevåg, Lilja Øvrelid, and Erik Velldal. 2020. NorNE: Annotating Named Entities for Norwegian. In *Proceedings of the 12th Edition of the Language Resources and Evaluation Conference*, Marseille, France, 2020.
- John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2020. Greek-Bert: The Greeks visiting Sesame street. In *11th Hellenic Conference on Artificial Intelligence*, pages 110–117.
- Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.
- Per E. Kummervold, Javier De la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. 2021. Operationalizing a National Digital Library: The Case for a Norwegian Transformer Model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics*.
- Robin Kurtz, Stephan Oepen, and Marco Kuhlmann. 2020. End-to-end negation resolution as graph parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 14–24, Online. Association for Computational Linguistics.
- Andrey Kutuzov and Elizaveta Kuzmenko. 2017. Building web-interfaces for vector semantic models with the WebVectors toolkit. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 99–103, Valencia, Spain. Association for Computational Linguistics.
- Petter Mæhlum, Jeremy Barnes, Robin Kurtz, Lilja Øvrelid, and Erik Velldal. 2021. Negation in Norwegian: an annotated dataset. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics*.
- Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the National Library of Sweden – making a Swedish BERT. *arXiv preprint arXiv:2007.01658*.
- Roser Morante and Eduardo Blanco. 2012. *SEM 2012 shared task: Resolving the scope and focus of negation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 265–274, Montréal, Canada.
- Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. Phobert: Pre-trained language models for vietnamese. *arXiv preprint arXiv:2003.00744*.
- Lilja Øvrelid and Petter Hohle. 2016. Universal Dependencies for Norwegian. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC’16)*, pages 1579–1585, Portorož, Slovenia.
- Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2020. A fine-grained sentiment dataset for Norwegian. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5025–5033, Marseille, France. European Language Resources Association.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pre-trained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepLanLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.
- Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta.
- Per Erik Solberg, Arne Skjærholt, Lilja Øvrelid, Kristin Hagen, and Janne Bondi Johannessen. 2014. The Norwegian Dependency Treebank. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, Reykjavik, Iceland.
- Matej Ulčar and Marko Robnik-Šikonja. 2020. High quality ELMo embeddings for seven less-resourced languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4731–4738, Marseille, France.
- Erik Velldal, Lilja Øvrelid, Eivind Alexander Bergem, Cathrine Stadsnes, Samia Touileb, and Fredrik Jørgensen. 2018. NoReC: The Norwegian Review Corpus. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference*, pages 4186–4191, Miyazaki, Japan.
- Erik Velldal, Lilja Øvrelid, and Petter Hohle. 2017. Joint UD parsing of Norwegian Bokmål and Nynorsk. In *Proceedings of the 21st Nordic Conference of Computational Linguistics*, pages 1–10, Gothenburg, Sweden.
- Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish. *arXiv preprint arXiv:1912.07076*.
- Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. Bertje: A Dutch Bert model. *arXiv preprint arXiv:1912.09582*.
- Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large batch optimization for deep learning: Training BERT in 76 minutes. In *International Conference on Learning Representations*.
- Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Extremely low-resource machine translation for closely related languages

Maali Tars, Andre Tättar, Mark Fišel

University of Tartu

{maali.tars, andre.tattar, mark.fishel}@ut.ee

Abstract

An effective method to improve extremely low-resource neural machine translation is multilingual training, which can be improved by leveraging monolingual data to create synthetic bilingual corpora using the back-translation method. This work focuses on closely related languages from the Uralic language family: from Estonian and Finnish geographical regions. We find that multilingual learning and synthetic corpora increase the translation quality in every language pair for which we have data. We show that transfer learning and fine-tuning are very effective for doing low-resource machine translation and achieve the best results. We collected new parallel data for Võro, North and South Saami and present first results of neural machine translation for these languages.

1 Introduction

Neural machine translation (NMT, Vaswani et al., 2017) shows great results in terms of output fluency and overall translation quality, however it relies on large parallel corpora for training the models. Low-resource NMT techniques like back-translation (Sennrich et al., 2016), multilingual knowledge transfer (Johnson et al., 2017; Ngo et al., 2020) and unsupervised NMT (Lample et al., 2018) rely on using parallel corpora for other languages and/or large quantities of monolingual data for the language(s) of interest.

Here we put these techniques to the test in an extremely low-resource setting, working on NMT systems for Võro-Estonian. While Estonian has plentiful parallel, monolingual and annotated corpora (Tiedemann, 2016; Nivre et al., 2020, etc), Võro with its 87 000 speakers and no normalized orthography only has slightly over 162 000 monolingual sentences with much less parallel data.

Here we resort to the help of languages closely related to Võro and Estonian: the resource-rich Finnish and two more extremely low-resource North and South Saami. We combine multilingual transfer learning, back-translation and then evaluate several combinations of these techniques on NMT for the five chosen Uralic languages.

Our contributions in this paper are as follows:

- experimental results for combinations of techniques for low-resource NMT with application to closely related resource-poor Uralic languages
- first developed NMT systems for Võro, North and South Saami languages with a free online demo¹
- additional data collected for Võro, North and South Saami

Next we review related work in Section 2, describe our experimental setup in Section 3, then proceed with results in Section 4 and conclude the paper in Section 5.

2 Related work

This section describes prior work in machine translation (MT) with neural networks for low-resource related languages. Our work on neural machine translation relies on (Vaswani et al., 2017), who introduce transformer, an encoder-decoder type of solution for MT based on self-attention.

2.1 Low-resource NMT

There has been a lot of research into low-resource MT, for example, phrase-based unsupervised and semi-supervised MT (Lample et al., 2018; Artetxe et al., 2018), but they relied on lexicons or large quantities of monolingual data. Their work is

¹<https://soome-ugri.neurotolge.ee/>

not easily applicable for our experiments because the amount of monolingual data is not sufficient, having less than 100K sentences for most of the languages in our data sets. The authors in (Hämäläinen and Alnajjar, 2019) used a template based approach to generate more parallel data for related languages, which made NMT models viable for training.

Another way of doing multilingual NMT is via zero-shot translations for very low resource language pairs. In our case, we have data for ten translation directions and zero parallel data for the rest of the ten directions. In (Gu et al., 2018) the authors showed that zero-shot translations achieve better results than the pivoting approach - pivoting means that when we have a language pair with sufficient data, then Võro to Finnish translation, which has zero data, would use the Estonian language to pivot - Võro to Estonian to Finnish translation. We want to avoid pivoting because Võro to North Saami would require two pivots or three translations in total, resulting in serious error propagation. Additionally, the authors use shared source embeddings and source RNN encoders; we used transformers with shared vocabulary, encoders and decoders. In (Rikters et al., 2018) the authors showed that multilingual training with transformers is optimal for multilingual Estonian-English-Russian system, but reported that high-resource pairs see a performance degradation and lower-resourced pairs see a performance increase.

2.2 Back-translation for low-resource MT

Every sentence is essential for neural machine translation in a low-resource machine translation environment. One popular way to leverage monolingual data is by creating a synthetic corpus via a method called back-translation (BT). Traditional BT (Sennrich et al., 2016) is easy to use and requires training a target-to-source MT system to generate translations of the monolingual data, which are used as training data for the source-to-target MT model. This means that traditional BT requires two NMT models, where one generates synthetic data for the other. The idea behind BT is that the monolingual human data on the target side improves the quality of the decoder to generate better output for the language and the synthetic source helps as a data augmentation tactic.

Closely related to back-translation is a method

called forward-translation (FT), where the model creates synthetic parallel data for itself - the source sentence is translated into the target language, and together, a bitext sample is created. In other words, forward-translation is called self-training. The authors in (Popović et al., 2020) used both BT and FT for closely related languages. They used a multilingual encoder (English and German) and a multilingual decoder (Serbian and Croatian) and achieved better results compared to single directional baselines in their experiments.

Our work is about a single multilingual system that enables the model to generate synthetic data for itself - both back-translation and forward-translation is used. The generated synthetic data is added to available parallel corpora as training data.

2.3 Transfer learning and fine-tuning

The authors in (Kocmi and Bojar, 2018) did trivial transfer learning for low resource NMT - in detail, they used a high resource language pair like English-Finnish to train a parent model. They continued training on a lower resource child model English-Estonian and showed that this improved translation quality significantly, 19.74 BLEU score compared to 17.03 when using only English to Estonian data. Additionally, they showed that “unrelated” languages might work even better, where the best English-Estonian results were achieved by using an English-Czech as a parent, which achieved a 20.41 BLEU score on the same test set. Their work shows that transfer learning is a very viable option for low-resource NMT. The only drawback is that their work still relies on some amount of data and a common source or target language to either share the encoder or decoder weights. In our case, there are language pairs, which have 0 available sentences like Võro to North Saami. Additionally, their work would require 20 such models to be trained.

The authors in (Currey and Heafield, 2019; Zhang et al., 2020) show that using multilingual back-translation for fine-tuning a multilingual model is beneficial for translation quality. Additionally, (Zhang et al., 2020) shows that their random online back-translation lowers the chance of the model doing off-target translations, which in our case is also a problem since the model never sees some language pairs. We build upon this work by doing two iterations of fine-tuning on a

synthetic back-translation corpora, where we uniformly at random assign the target language into which to translate.

The difference between transfer learning and fine-tuning is small. We refer to transfer learning when the MT model is trained on some languages that the model has never seen before, e.g., when using ET-FI model weights to initialize the ET-VRO model. We refer to fine-tuning when we continue training a multilingual model on data, which the model has seen before, e.g., when using the multilingual model to fine-tune on ET-VRO data only.

2.4 Source factors for multilingual zero-shot NMT

We rely on the work of (Sennrich and Haddow, 2016) for zero-shot translations in our multilingual models; in that article, the authors use morphological features like POS tags to enrich source-side representations. We use source-side factors to give the transformer model information about the intended target language, so the model knows which language the output should be in. The authors in (Tars and Fishel, 2018) used source-factors to give domain and target language information for the model. Using source factors is similar to using a single token on the input sentence to distinguish between closely related languages and dialects (Lakew et al., 2018; Costa-jussà et al., 2018), where authors show an improvement over a single baseline model when training a model for similar languages.

3 Experimental setup

3.1 Data sets

3.1.1 Preprocessing

The data for the experiments originated from many different sources. Subsequently, the main issue with the parallel data collected was the differences in file formats, which took a long time to solve in order to create a unified data set. The biggest problem with parallel data was that there were a lot of repeated sentence pairs in the data, which required a uniqueness check and reduced the number of sentence pairs for the Finnish-North Saami (FI-SME) language pair by about 75 percent, as seen in Table 1.

Preprocessing monolingual data was also a long process as there were no conclusive ready-made sets available for languages like Võro, North

Saami and South Saami. As described in Table 2, in the first set, the data consisted mostly of news corpuses, fiction and Wikipedia texts. The data files were in different formats, as was the case with parallel data. Estonian and Võro required extracting sentences from texts and removing empty lines. The Võro, North Saami and South Saami data in the second set was gathered manually from news articles and various PDF style documents (fiction, scientific texts, official documents) available. The paragraphs of text then needed to be divided into sentences and joined into one TXT type file for compatibility. Additional preprocessing included fixing some minor alignment issues.

3.1.2 Validation and test data

Validation and test sets consisted of sentences from all the five language pairs mentioned in Table 1. The number of sentences for each language pair was chosen proportionally to the amount of training data the pair had. In total, there were 1862 test sentences and 939 validation sentences. There is no official test set available for these language pairs collectively, and as parallel data was scarce, the validation and test sets were sentences that were randomly held-out of the training data.

3.1.3 Parallel data

Table 1 also highlights the fact that Estonian-Finnish (ET-FI) acted as the high-resource language pair in the experiments, with 2.6 million sentence pairs available. Other language pairs formed a small fraction of the whole parallel data set, with about 1 percent. The lowest amount of data was discovered for Finnish-South Saami (FI-SMA) language pair, with under 3000 sentence pairs.

3.1.4 Monolingual data

As expected, Estonian and Finnish had the most monolingual data available. Although finding data sets for the low-resource languages proved to be more difficult, there was more of it available than parallel data for their respective language pairs used in this work. The two sets of monolingual data described in Table 2 were collected separately. Experiments with the first set were already performed prior to gathering the second set, which is why the amounts of two sets are off-balance. For the sake of the models learning more about low-resource languages, we used the down-sampling technique, reducing the amount of Esto-

Language pair	Before cleaning	After cleaning	Eliminated
et-fi (Tiedemann, 2016)	3 566 826	2 646 922	919 904
et-vro ²	30 816	30 502	314
fi-sme ³	109 852	35 426	74 426
fi-sma ³	3098	2895	203
sme-sma ³	23 746	21 557	2189
Overall	3 734 338	2 737 302	997 036

Table 1: Parallel data sets (in sentence pairs). et - Estonian, fi - Finnish, vro - Võro, sme - North Saami, sma - South Saami.

nian and Finnish monolingual data to level them with the amount of low-resource language monolingual data in use. That made Võro (VRO) the most prominent language in the data set, as shown in Table 2. The amount of data for North and South Saami was still quite low, but it was an improvement over the parallel data set numbers.

3.2 Models and parameters

3.2.1 General settings

In our experiments we use the Sockeye framework described by (Hieber et al., 2017), which has implemented source-side factors where we give the target language token as an input feature for the transformer model. During training, the vocabulary that was created included all of the languages. Specifications of the training process included setting the batch size to 6000 words and checkpoint interval to 2000. All the models in the experiments trained until 32 consecutive unimproved checkpoints were reached. The unimproved metric was perplexity. All of the experiments use the standard transformer parameters (6 encoder and 6 decoder layers with 8 attention heads and size 512). Prior to training, all of the data used to develop the models was tokenized by a SentencePiece (Kudo and Richardson, 2018) tokenization model, which follows the byte-pair encoding algorithm. The tokenization model was previously trained on all of the training data.

²<https://doi.org/10.15155/1-00-0000-0000-0000-001A0L>

³<https://giellalt.uit.no/tm/TranslationMemory.html>

⁴<https://www.cl.ut.ee/korpused/segakorpus/epl/>

⁵<https://doi.org/10.15155/1-00-0000-0000-0000-00186L>

⁶<https://github.com/maalitars/FinnoUgricData>

⁷<http://hdl.handle.net/11509/102>

3.2.2 Multilingual baseline

One of the fundamental experiments of this work was developing the multilingual baseline model, which had five source languages and five target languages. This means that this model could produce translations in 20 different directions. For this, each pair of parallel data seen in Table 1 was copied and the source-target direction was switched. The turned-around parallel data set was then added to the original data set and the multilingual baseline model was trained on all of the combined parallel data. The data set was tokenized by a tokenization model, which was trained on all the training data from the parallel data set, meaning text patterns were generalized over five languages.

3.2.3 Back-translation experiments

Synthetic parallel data via back-translation was produced in two iterations and additional models were also trained in two iterations. The monolingual data was translated into every other language in equal measures. For example, 1/4 of the 100 000 sentences in Estonian were translated into Finnish, 1/4 into Võro, 1/4 into North Saami and 1/4 into South Saami. The paired-up synthetic translations and monolingual data made up the additional parallel data corpus.

Combining the new synthetic parallel data corpus and the original, human-translated corpus, gives the models more parallel data to learn on during training. The methodology of both back-translation data experiment iterations was the same, but there were some important aspects that were different:

First iteration

- Monolingual data used: first monolingual data set
- The first batch of synthetic data was produced with the multilingual baseline model. The

Language	First set	Second set	All
et ⁴	100 000	25 000	125 000
fi (Goldhahn et al., 2012)	100 000	25 000	125 000
vro ^{5,6}	162 807	5290	168 097
sme (Goldhahn et al., 2012; Tiedemann, 2012), ⁶	33 964	6057	40 021
sma ^{6,7} (Tiedemann, 2012),	55 088	5377	60 465

Table 2: Monolingual data sets after preliminary cleaning (in sentences). et - Estonian, fi - Finnish, vro - Võro, sme - North Saami, sma - South Saami.

Model	et-fi	fi-et	et-vro	vro-et	fi-sme	sme-fi	fi-sma	sma-fi	sme-sma	sma-sme	$BLEU_{low}$
Baselines	32.0	29.4	14.6	17.5	28.0	28.7	4.6	6.3	8.3	9.1	14.6
Multilingual (ML)	30.9	29.5	23.8	29.6	31.3	34.7	9.4	9.4	19.8	19.8	22.2
+ BT1	32.4	29.9	25.2	29.4	32.3	36.1	10.8	9.9	20.3	20.0	23.0
+ BT1(*)	30.1	29.1	24.5	30.3	32.3	36.2	11.1	10.5	21.4	20.0	23.3
+ BT1 + FT1	31.3	30.1	25.2	31.5	31.3	35.7	8.9	10.0	18.7	20.4	22.7
+ BT1 + FT1(*)	30.9	28.8	25.8	30.4	31.5	35.7	8.9	10.1	19.4	20.1	22.7
+ BT2	31.5	30.2	26.0	31.0	32.3	36.6	11.3	10.9	20.3	21.0	23.7
+ BT1 + BT2(*)	31.3	29.6	26.2	31.3	31.4	36.4	12.4	10.6	21.6	20.7	23.8
+ BT1 + BT2(**)	30.4	29.7	25.1	31.6	31.7	37.5	11.4	10.3	21.3	20.9	23.7
+ BT1&2 + FT1&2(*)	30.2	29.4	25.1	31.7	31.5	36.8	9.5	9.7	20.4	20.6	23.2
BT1	21.1	21.6	20.5	24.9	24.0	27.4	8.5	7.3	15.9	14.5	17.9
BT1(*)	8.4	8.6	18.7	19.9	11.8	13.4	6.9	5.3	12.9	9.2	12.3

Table 3: BLEU scores. (*) - trained without pre-trained weights, (**) - trained on + *BT1*(*) weights. *BT* - back-translation data set, *FT* - forward-translation data set, $BLEU_{low}$ - average BLEU score on low-resource language pairs (excluding ET-FI and FI-ET), **bold** - best BLEU score for a language pair.

synthetic data was then added to the original parallel data and the training process was repeated, which produced a new model.

Second iteration

- Monolingual data used in this iteration consisted of 1) shuffled first monolingual data set, 2) second monolingual data set.
- Monolingual data was translated by the newest model that had been trained on parallel data and synthetic data from the first iteration of back-translation (+*BT1* in Table 3). Subsequently, a new model was trained using original parallel data plus the two batches of synthetic data produced.

Additional experiments included having different combinations of back-translation/forward-translation data and differences in initialized weights, with the best of them presented in Table 3.

3.2.4 Transfer learning experiments

We performed an experiment fine-tuning the multilingual baseline model on ET-VRO parallel data

and a transfer learning experiment, initializing ET-VRO model with ET-FI baseline model weights. Then we compared the results of these two experiments to each other and to the ET-VRO baseline model. The ET-VRO data was the same parallel data that was used for training the multilingual baseline model (*ML*).

4 Results

4.1 Quantitative analysis

Quantitative results were determined by comparing BLEU scores (Papineni et al., 2002), using the SacreBLEU implementation (Post, 2018) of calculating the score on detokenized sentences⁸. Multiple experiments were assessed and the best experiments are explained in Table 3. Additional analysis was done with the CHRF metric, which compares sentences on a character-level (Popović, 2015). We used the SacreBLEU implementation (Post, 2018) of the CHRF metric⁹ and the results can be seen in the Appendix in Table 7.

4.1.1 BLEU

Multilingual baseline. All of the low-resource language pairs experienced a positive gain over

baseline model results in comparison to the multilingual baseline model (*ML*) experiment. An average gain of 7.6 BLEU was achieved on the low-resource language pairs with VRO-ET and SME-SMA exceeding this average gain by an additional 4 BLEU points. Noticeably, FI-SMA made the smallest improvement, perhaps the main reason for this lies in FI-SMA having significantly less parallel data than other low-resource language pairs.

Back-translation experiments. Experiments with data from back-translation iterations further improved the BLEU score for low-resource language pairs compared to the multilingual baseline model. None of the models showed uniform improvements across all of the low-resource language pairs, however we can highlight one model with the highest average gain over baseline results, improving by +9.2 BLEU points. This model was trained on parallel data plus two batches of back-translation data but without any initialized weights (+ *BT1* + *BT2*(*)) in Table 3).

While the pre-trained weights did not seem to help produce the best models with parallel and back-translation data, the experiments with only back-translation data show that initializing a model with useful pre-trained weights can still be very helpful in the case of related tasks. This is illustrated by models *BT1* and *BT1*(*) with 7.1 BLEU points between them.

Experiments with added forward-translations did not appear to improve the results except for the VRO-ET language pair.

Transfer learning experiments. Transfer learning and fine-tuning a model for a particular language pair results in further improvements over the best back-translation model results. In this part, we performed two experiments. In the transfer learning experiment, we trained an ET-FI baseline model until convergence; then the training data was changed to the ET-VRO data set, which was used for training until convergence. In the second experiment, we fine-tuned the multilingual baseline model with the ET-VRO language direction data only. Comparing BLEU results in Table 5, it is clear that doing transfer learning for

low-resource NMT is very beneficial - a 12 BLEU point increase is achieved by doing trivial transfer learning, and even better gains are seen in the multilingual fine-tuning experiment with a 13 point BLEU score increase.

Thus, the best results were achieved in the transfer learning and the fine-tuning experiment. Transfer learning alone, however, has a downside. Compared to multilingual models, which can translate in 20 different directions, in case of transfer learning, to achieve the same functionality, 20 separate models would have to be trained, which takes up a lot more resources.

4.1.2 CHRF

For the low-resource language pairs, the CHRF score metric mostly agreed with the BLEU score metric on which model gives the best results for each language pair, except for SMA-FI and SMA-SME. This can be seen in the Appendix in Table 7. With the CHRF score, however, it is much clearer that the model + *BT1* + *BT2*(*) is the best one out of all the experiments done with back-translations, because both $BLEU_{low}$ and $CHRF_{low}$ had the best scores on test data with this model and six out of the eight low-resource language pairs achieved the highest CHRF scores. In Table 5, for transfer learning and fine-tuning experiments, the BLEU and CHRF scores moved in the same direction, transfer learning and fine-tuning improving results substantially.

Overall, we can see the same patterns, both in the BLEU and the CHRF score analysis: the multilingual model concept helps get better translation quality for low-resource languages compared to baseline results; adding more and more back-translated data to the training data increases the scores; adding forward-translations, however, mostly lowers the scores. Another noticeable thing shown by both of the scores, is that back-translation on its own, when looking at the models *BT1* and *BT1*(*), does not achieve good (and comparable) results. One possible reason for this is the data domain mismatch between test data (parallel data hold-out) and monolingual data. A balanced test set for these languages could provide a better overview of the results and give more accurate info on the quality of the models.

⁸SacreBLEU signature: BLEU+case.mixed+lang.LANG-LANG+numrefs.1+smooth.exp+test.SET+tok.13a+version.1.4.14 where LANG in {et,fi,vro,sme,sma}

⁹SacreBLEU signature: chrF2+lang.LANG-LANG+numchars.6+numrefs.1+space.false+test.SET+version.1.5.1 where LANG in {et,fi,vro,sme,sma}

a)	Source	Uue nime väljamõtlemisel oli tähtis, et oleks selge side kohaliku kogukonnaga ja et nimi aitaks jutustada ettevõtte lugu.
	Baseline	Vahtsõ opimatõrjaali saamisõs oll tähtsä, et tähtsä olõs ka selge sõnumiga tõsitsit luulõtuisi.
	ML	Vahtsõ nime vällämõtlemisel oll tähtsä, et olõsi selge side paigapäälitse kogokunnaga ja et nimi avitas kõnõlda ettevõtte lugu.
	+BT1+BT2(*)	Vahtsõ nime vällämõtõldõn oll' tähtsä, et olõs selge side paigapäälidse kogokunnaga ja et nimi avitas kõnõlda ettevõttõ lugu.
	Reference	Vahtsõ nime vällämärkmise man oll' tähtsä, et olõs selge köüdüs paikliku kogokunnaga ja et nimi avitanuq jutustaq ettevõtmisõ luku.
	<i>English</i>	On coming up with a new name, it was important that there was a clear reference to the local community and that the name would help tell the story of the business.
b)	Source	Parhilla ommaq jutuq hindamiskogo käen, kokkovõtõq ja preemiäsaajaq trükitäseq ärq järgmädsen Uman Lehen .
	Baseline	Praegu on instruksioone, ka kokku võtted , selliste sündmuste ja aegajalt „sisse lülitada” kaugemate Leivalentsemad.
	ML	Praegu on jutud hindamiskogu käes, kokku võtted ja preemiasaajad trükivad järgmise Uman Leheni .
	+BT1&2+FT1&2(*)	Praegu on jutud hindamiskogu käes , kokku võtted ja preemiasaajad trükivad ära järgmises Uman Lehes .
	Reference	Praegu on jutud hindamiskomisjoni käes, kokku võte ja preemiasaajad trükitakse ära järgmises Uma Lehes .
	<i>English</i>	At the moment the stories are with the judging committee, the summary and the winners will be printed in the next Uma Leht .
c)	Source	Nuoria on tullut tilalle aika lailla , ehkä ottaa eräs nuori kyläelämän vetämisen haltuunsa.
	Baseline	Nuorat leat bohtán sadjái áigi, soadi, kántorin jos čadahat gilvun.
	ML	Nuorat leat bohtán sadjái áiggi ládje , soaitá váldit ovtta nuorra gilieallima jodiheami.
	+BT1	Nuorat leat bohtán sadjái áige ládje , soaitá váldit ovtta nuorra gilieallima jodiheami háldui.
	Reference	Nuorat lea bohtán lasi oalleláhkái , gánske muhtun nuorra váldá gilieallima geassima iežas háldui.
	<i>English</i>	There are quite a bit more younger people now, maybe one of them will take over the role of leading the village life.

Table 4: Example translations from a) ET-VRO, b) VRO-ET and c) FI-SME.

Model	BLEU	CHRf
ET-VRO baseline	14.6	0.393
ET-VRO on ET-FI weights	26.5	0.540
ML fine-tuned on ET-VRO	27.6	0.563

Table 5: BLEU and CHRf scores for transfer learning and fine-tuning experiments.

4.2 Qualitative analysis

Table 4 compares some sentence translations for ET-VRO, VRO-ET and FI-SME language pairs. It is clear that baseline models produced subpar translations, rendering them non-sensical. The multilingual baseline model improved the translations significantly, although still making some detrimental mistakes, like choosing the wrong word, so the meaning is lost, or deciding not to translate some parts of the sentence. Adding back-translation data to the models fixed some of these mistakes and made some important changes in understanding the meaning of a sentence, but the best back-translation models still left in some grammatical errors, such as wrong verb forms, grammatical cases and tenses.

In the first example, translating in the ET-VRO direction, the best model chooses a direct translation of “väljamõtlemisel” to “vällämõtöldõn”. In addition, all of the models omit the “q” endings of the words “avitanaq” and “jutustaq” or “kõnõldaq”. This symbol usually signifies plurality and, in a lot of cases upon translating in the ET-VRO direction, the models chose not to add the “q”, although it would have been correct. The problem could lie in the data, where the “q” endings are also not always added, which in turn could confuse the models.

The second example illustrates a VRO-ET direction translation, which presents some bigger flaws. For example, handling names is difficult even for the best model, with “Uman Lehen” being translated incorrectly to “Uman Lehes”. The grammatical case of the word “Lehen” was correctly changed to have an “s” ending, but the case of the word “Uman” was not changed. In addition, “trükitäseq” is translated to the wrong verb form “-vad”, but it should be replaced with the impersonal form “-takse”. Continuing with the problems caused by the “q” ending, here the word “kokkovõtõq” is translated to plurality “kokkuvõtted”, but in this particular case, it should be translated to the singular form “kokkuvõte”.

The third example shows the FI-SME language pair translations. Here the meaning of the sentence is understandable, but the word-pair “áige ládje” is a direct translation from the Finnish phrase “aika lailla”, which means “quite a bit” in English, but “áige ládje” does not hold the same meaning.

Additional examples can be seen in the Appendix in Table 6. In these examples, there is another flaw presented, which might be unique to multilingual models, where some words in a sentence are translated into the wrong language, although they might have the correct meaning. This is illustrated very well in the third ET-VRO translation example in Table 6. All models, except the baseline, choose to translate the word “ametlikult” into the Finnish word “virallisesti”, instead of trying to find a word for it in Võro language.

4.3 Discussion

The results show that synthetic data helps to learn a better model; however, the model which has continued training on only back-translation data sees performance degradation. This is most likely caused by the test sets’ domain mismatch problem and is alleviated by merging the parallel and synthetic data into one big corpus. This shows that a new separate test set should be created for this problem, but it is very hard to do as there are very few speakers. We have started to gather a multilingual five-way test corpus.

We think that this work can be further improved by doing better multilingual fine-tuning, shown by promising multilingual fine-tuning experiments, where the best result was 27.6 BLEU points compared to the best multilingual model, which achieved 26.2 BLEU points. The research question remains, how well would fine-tuning work for a completely synthetic parallel corpus like VRO-SMA. We did not explore this yet due to not having enough resources. Also, it is unknown what effect this single language pair fine-tuning would have on other languages.

5 Conclusions and future work

Multilingual neural machine translation with shared encoders and decoders work very well for very low resource language translation. Using back-translation for low resource MT is vital for best results, further improved by transfer learning and fine-tuning.

In the future, we hope to work with more Uralic

languages and add an unrelated high-resource language, for example German. Secondly, we want to do better multilingual fine-tuning since the best ET-VRO score of 27.6 was reached by multilingual fine-tuning, compared to 26.2 for multilingual training. Finally, we hope to find more parallel and monolingual data.

References

- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.
- Marta R Costa-jussà, Marcos Zampieri, and Santanu Pal. 2018. A Neural Approach to Language Variety Translation. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 275–282, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Anna Currey and Kenneth Heafield. 2019. Zero-resource neural machine translation with monolingual pivot data. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 99–107, Hong Kong. Association for Computational Linguistics.
- D. Goldhahn, T. Eckart, and U. Quasthoff. 2012. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the 8th International Language Resources and Evaluation (LREC'12)*.
- Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana. Association for Computational Linguistics.
- Mika Härmäläinen and Khalid Alnajjar. 2019. A template based approach for training nmt for low-resource uralic languages - a pilot with finnish. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence, ACAI 2019*, page 520–525, New York, NY, USA. Association for Computing Machinery.
- Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium. Association for Computational Linguistics.
- Taku Kudo and John Richardson. 2018. Sentence-Piece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Surafel Melaku Lakew, Aliia Erofeeva, and Marcello Federico. 2018. Neural Machine Translation into Language Varieties. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 156–164, Brussels, Belgium. Association for Computational Linguistics.
- Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.
- Thi-Vinh Ngo, Phuong-Thai Nguyen, Thanh-Le Ha, Khac-Quy Dinh, and Le-Minh Nguyen. 2020. Improving multilingual neural machine translation for low-resource languages: French, English - Vietnamese. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 55–61, Suzhou, China. Association for Computational Linguistics.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the*

- Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.
- Maja Popović, Alberto Poncelas, Marija Brkic, and Andy Way. 2020. Neural Machine Translation for translating into Croatian and Serbian. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 102–113, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).
- Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Mat Rikters, Mārcis Pinnis, and Rihards Krišlauks. 2018. Training and Adapting Multilingual NMT for Less-resourced and Morphologically Rich Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- Sander Tars and M. Fishel. 2018. Multi-domain neural machine translation. *ArXiv*, abs/1805.02282.
- Jörg Tiedemann. 2016. OPUS – parallel corpora for everyone. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation: Projects/Products*, Riga, Latvia. Baltic Journal of Modern Computing.
- Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12)*, Istanbul, Turkey. European Language Resources Association (ELRA).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.
- Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In
- Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Appendix

A

Source	Ja ühe hea külje leiab siidritegija uue nime juures veel.
Baseline	Ja üte hää ütest põlvost inemiisist lõpõ nime man vahtsõ nime man.
ML	Ja üte hää küle om siidritegijä vahtsõ nime man viil.
+BT1+FT1(*)	Ja üte hää küle löüid siidritegijä vahtsõ nime man viil.
+BT1+BT2(*)	Ja üte hää küle lövvüis siidritegijä vahtsõ nime man viil.
Reference	Ja üte hää küle löüid siidritegijä vahtsõ nime man viil.

Source	Samas tegutsevad kotkad kultuurmaastikul, mis tähendab, et ka inimesel on tähtis roll selles, et neil hästi läheks.
Baseline	Samal aol om mi kotustõ perändüskultuurmaastikkõ, miä tähendäs, et inemisel tähtsä om tähtsä, et näil olõ-i tähtsä.
ML	Saman omma' kotka' kultuurmaastikul, miä tähendäs, et ka inemisel om tähtsä roll tan, et näil häste lääsi.
+BT1+FT1(*)	Samal aol omma kotka kultuurmaastikul, miä tähendäs, et ka inemisel om tähtsä roll tuun, et näil häste lääsi.
+BT1+BT2(*)	Saman toimõndasõq kotkaq kultuurmaastikul, miä tähendäs, et ka inemisel om tähtsä roll tuun, et näil häste lääsiq.
Reference	Saman toimõndasõq kotkaq kultuurmaastikul, miä tähendäs, et ka inemisel om tähtsä roll tuu man, et näil häste lännüq.

Source	Leevakul elab ametlikult pea 300 inimest.
Baseline	Leevälapjo eläs tähtsä pää inemist.
ML	Leväkul eläs virallisesti pää 300 inemist.
+BT1+FT1(*)	Leevakul eläs virallisesti pää 300 inemist.
+BT1+BT2(*)	Leevakul eläs virallisesti pia 300 inemist.
Reference	Leevakul eläs kirjo perrä pia 300 inemist.

B

Source	A edesi ei näeq tükk aigo tii pääl üttegi võrokiilset silti.
Baseline	Aga edasi ei näinud tükk aega, tee üttegi saaklooma ära.
ML	Aga edasi ei näe tükk aega tee peal üttegi võrukeelset ikka.
+BT1+FT1	Aga edasi ei näe tükk aega tee peal üttegi võrukeelset silti.
+BT1&2+FT1&2(*)	Aga edasi ei näe tükk aega teel üttegi võrukeelset silti.
Reference	Aga edasi ei näe tee peal tükk aega üttegi võrukeelset silti.

Source	Ütelt puult tulõ hoita vannu mõtsu, et nä saanu ummi pesi kohegi ehitä.
Baseline	Ühis poolt tuleb hoida vanade metsade, et nad saanud märkimisväärsset kuhugi ehitatud.
ML	Ühel pool tuleb hoida vana metsa, et nad saaksid oma pesu ehitada.
+BT1+FT1	Ühel pool tuleb hoida vana metsa, et nad saaksid oma pesi kuhugi ehitada.
+BT1&2+FT1&2(*)	ühelt poolt tuleb hoida vanu metsasid, et nad saaksid oma pesi kuhugi ehitada.
Reference	Ühelt poolt tuleb hoida vanu metsi, et nad saaks oma pesasid kuhugi ehitada.

Source	Nii om võimalus telefon võita ka Uma Lehe teljal .
Baseline	Nii on võimalus telefongu ka Uma Pidoga mitmeti seotud.
ML	Nii on võimalus telefon võita ka Uma Lehe telgis .
+BT1+FT1	Nii on võimalus telefon võita ka Uma Lehe telgil .
+BT1&2+FT1&2(*)	Nii on võimalus telefon võita ka Uma Lehe telgil .
Reference	Nii on võimalus telefon võita ka Uma Lehe tellijal .

C

Source	Uutta nimeä keksiessä oli tärkeää, että olisi selkeä yhteys paikalliseen yhteisöön ja että nimi auttaisi kertomaan yrityksen tarinan.
Baseline	M uhccin muitalin lei dehålaš, ahte livččii čielga oktavuoha båkåålaš servodahkii ja ahte namma veahkehivččii muitalit lihkastagaide.
ML	Odđa namma lei dehålaš, ahte livččii čielga oktavuoha båkåålaš servošii ja ahte namma veahkehivččii muitalit fitnodaga máidnasiid.
+BT1	Odđa nama huksemis lei dehålaš, ahte livččii čielga oktavuoha båkåålaš servodahkii ja ahte namma veahkehivččii muitalit fitnodaga máidnasa.
Reference	Odđa nama hutkkadettiin lea dehålaš, ahte livčče čielga oktavuoha båkåålaš servodahkii ja ahte namma veahkehivčče muitalit fitnodaga muitalusa.

Source	Kansa kokoontuu entiseen koulutaloon , jossa on myös kirjasto.
Baseline	Riikkabeavevahku čoačkkanå sågadoalliiriikkas, mas leat maid girjerájus.
ML	Álbmoga čoačkkanå ovddeš skuvllas , mas lea maid girjerádju.
+BT1	Álbmot čoačkkanå ovddeš skuvladássái , mas lea maid girjerádju.
Reference	Álbmot čoačkkanå boares skuvlavistái , mas lea maidái girjerádju.

Source	Sosiaalissa mediassa pitivät ihmiset eniten tehtävästä "Puhu tai postaa yksi viesti tai tarina võron kielellä ".
Baseline	Sosiåla medias atne olbmot eanemus barggus " Ominayak oktavuodavåldimiid dehe máidnumaõjstõõllåmkerjij lea oaivvilduvvon.
ML	Sosiåla medias doalai olbmuid eanemus bargguin "Puhu dahje poasta okta nja dahje máidnasa gillii ."
+BT1	Sosiåla medias dolle olbmot eanemusat bargguin "Puhu dahje poasta okta njaš dahje máidnasa vuonagillii ."
Reference	Sosiåla medias liikojedje olbmot maiddái bargobihåas "Muital dahje poste ovtta cukcasa dahje máidnasa võro gillii ."

Table 6: Translation examples. A: Estonian-Võro, B: Võro-Estonian, C: Finnish-North Saami. **blue** - incorrect word, **violet** - incorrect form/case/tense or partially incorrect.

Model	et-fi	fi-et	et-vro	vro-et	fi-sme	sme-fi	fi-sma	sma-fi	sme-sma	sma-sme	$CHRF_{low}$
Baselines	0.602	0.573	0.353	0.390	0.577	0.577	0.282	0.274	0.330	0.301	0.385
Multilingual (ML)	0.595	0.578	0.510	0.540	0.631	0.650	0.376	0.348	0.546	0.525	0.516
+ BT1	0.600	0.583	0.531	0.551	0.639	0.659	0.408	0.348	0.557	0.531	0.528
+ BT1(*)	0.592	0.575	0.526	0.556	0.639	0.659	0.420	0.349	0.566	0.532	0.531
+ BT1 + FT1	0.595	0.584	0.526	0.558	0.634	0.654	0.369	0.353	0.544	0.533	0.525
+ BT1 + FT1(*)	0.596	0.575	0.537	0.558	0.636	0.656	0.392	0.349	0.551	0.531	0.525
+ BT2	0.598	0.585	0.535	0.560	0.640	0.663	0.418	0.358	0.563	0.537	0.534
+ BT1 + BT2(*)	0.595	0.583	0.539	0.565	0.636	0.663	0.436	0.364	0.569	0.539	0.539
+ BT1 + BT2(**)	0.594	0.579	0.530	0.563	0.640	0.665	0.423	0.354	0.567	0.536	0.535
+ BT1&2 + FT1&2(*)	0.592	0.578	0.530	0.565	0.634	0.662	0.399	0.350	0.564	0.539	0.530
BT1	0.526	0.515	0.480	0.523	0.582	0.607	0.371	0.296	0.524	0.488	0.484
BT1(*)	0.349	0.356	0.455	0.473	0.459	0.443	0.348	0.253	0.488	0.402	0.415

Table 7: CHRF scores. (*) - trained without pre-trained weights, (**) - trained on + $BT1(*)$ weights. BT - back-translation data set, FT - forward-translation data set, $CHRF_{low}$ - average CHRF score on low-resource language pairs (excluding ET-FI and FI-ET), **bold** - best CHRF score for a language pair.

Measuring Translationese across Levels of Expertise: Are Professionals more Surprising than Students?

Yuri Bizzoni* Ekaterina Lapshinova-Koltunski*

Saarland University, Campus A2.2, Saarbrücken, Germany

yuri.bizzoni@uni-saarland.de

e.lapshinova@mx.uni-saarland.de

Abstract

The present paper deals with a computational analysis of translationese in professional and student English-to-German translations belonging to different registers. Building upon an information-theoretical approach, we test translation conformity to source and target language in terms of a neural language model’s perplexity over Part of Speech (PoS) sequences. Our primary focus is on register diversification vs. convergence, reflected in the use of constructions with a higher vs. lower perplexity score. Our results show that, against our expectations, professional translations elicit higher perplexity scores from the target language model than students’ translations. An analysis of the distribution of PoS patterns across registers shows that this apparent paradox is the effect of higher stylistic diversification and register sensitivity in professional translations. Our results contribute to the understanding of human translationese and shed light on the variation in texts generated by different translators, which is valuable for translation studies, multilingual language processing, and machine translation.

1 Introduction

Translationese is a set of linguistic patterns that tell translations apart from texts originally written in the same language and that make translations stylistically more similar to each other than original texts tend to be. While translationese was extensively discussed in the area of corpus-based translation

studies and machine translation (MT) (Zhang and Toral, 2019; Graham et al., 2020, among others), there are relatively few computational studies that focus on the varying amount of translationese characterizing different kinds of written translation (see Section 2.2 below). This study focuses on the relation between translators’ level of expertise and translationese throughout different registers. If we can connect translationese at least partly to the translator’s experience, we can expect to find different degrees of translationese between student and professional translations. As translationese is probabilistic in nature (Toury, 2004), we use a framework that enables a probabilistic design of language use in the form of a language model. We model language conventions in terms of grammatical structures represented by PoS sequences through Long Short-Term Memory (LSTM), a recurrent neural network architecture, using monolingual corpora of non-translations in both source and target language as a training set. We then test how students’ and professionals’ translations conform to linguistic conventions using our models’ perplexity scores. Through this approach, we aim at testing two related hypotheses:

Hypothesis 1 Overall, we can expect professional translators to be more efficient at reproducing the patterns of their target language. If this is the case, we would expect professional translations to elicit lower perplexity scores from the target language model than from the model of the source language.

Hypothesis 2 On the other hand, it is possible that students converge more on standard patterns: due to their lack of expertise, they might have lower register sensitivity, and thus they could be less bold and more repetitive in their use of grammatical constructions. A higher value of perplexity for a register means a less usual (hence, more perplexing) order of PoS with respect to a reference corpus,

*Both authors contributed equally.

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: <http://creativecommons.org/licenses/by/4.0/>.

and hence a more distinct register. So a higher level of convergence would result in more homogeneous surprisal values across registers.

We then compare the results of our perplexity measures with the distribution of different PoS patterns across registers to qualitatively analyze translation divergence in the data.

We organized the remainder of the paper as follows: Section 2 introduces the main concepts we are developing our research on and provides an overview of the related work, Section 3 includes details on the data and methods used in the analyses. We show the results of the analyses in Section 4 and 5, interpret them, and conclude in Section 6.

2 Main Concepts and Related Work

2.1 Distinctive features of translations

As we mentioned above, translationese is related to a set of distinctive linguistic features that make translations differ from non-translations (Gellerstam, 1986; Baker, 1996; Toury, 1995). Translationese appears to be a ubiquitous phenomenon, and happens in different forms both in human and machine translations (Graham et al., 2020; Zhang and Toral, 2019; Bizzoni et al., 2020). Automatic classifications of texts into translations and non-translations usually operationalize translationese as a combination of lexico-grammatical and textual features of different kinds (Baroni and Bernardini, 2006; Laippala et al., 2015; Volansky et al., 2015; Rabinovich et al., 2017). The number of such features, as well as their designation, varies across translation studies. The following macro-categories are relevant for our study: **shining through** – translations reproducing patterns typical of the source language instead of following the target language conventions (Teich, 2003)¹; **normalization** – translations conforming to patterns and practices which are typical of the target language (Baker, 1996), and **convergence** – the tendency of translated language to be more homogeneous in terms of the distribution of language patterns (Laviosa, 2002). In our definition of convergence, we follow the study by Kruger and van Rooy (2012) who analyze it across registers and conceptualize it as a form of register insensitivity. The main idea is that translations show less variation, which reduces the distinctness of vari-

¹Shining through is related to the law of interference, according to which phenomena of the make-up of the source text tend to get transferred to the target text (Toury, 1995).

ous registers. While in this sense, there might be no “perfect” translation (no translation completely indistinguishable from comparable originals), we are interested in the degree to which professional and non-professional translators are sensitive to register.

We can observe translationese at the lexical level, i.e., displaying less lexical and semantic diversity than the original (Baroni and Bernardini, 2006; Bizzoni and Teich, 2019), and at the grammatical level, i.e., using more typical syntactic construction instead of unusual ones (Ilisei et al., 2010; Volansky et al., 2015). In our analyses, we test translation conformity with either the target or the source language through language models’ perplexity scores measured over PoS sequences which represent grammatical level.

2.2 Translationese and translation expertise

Computational analyses of professional and novice translations are based on the assumption that translations of different levels of expertise manifest translationese to different degrees. Redelinguys (2016) compare non-translated English texts with translations by experienced and inexperienced translators in terms of frequencies of features like conjunctive markers, standardized type-token ratio, and word length, performing a univariate analysis for individual features. Kunilovskaya and Lapshinova-Koltunski (2020) report on two separated translationese effects and find a correlation between the levels of expertise and types of the detected effects. However, they ignore register differences. Lapshinova-Koltunski (2020) shows in her analyses of the same translation dataset we are using in our work that there are register-specific effects on the normalization and shining through for professional and student translations. Her results are based on such measures as distribution of content and grammatical words, nominal and verbal categories, various types of pronouns a.o., however, and do not provide any significant differences between student and professional translators. Corpas Pastor et al. (2008) and Ilisei (2012) use supervised machine learning techniques to distinguish between non-translations in Spanish and English-Spanish translations by professionals and students, investigating the validity of the translation universal of convergence. However, their definition of convergence is different from ours – they define this as the similarity between texts trans-

lated by translators of different proficiency levels and do not find significant differences between student and professional translations in terms of the features applied. We relate our analysis to the study by Martínez and Teich (2017) who also apply a probabilistic approach to study differences in the lexical choices by professional and student translators related to source-dependent and target-dependent translationese. Rubino et al. (2016) also use surprisal measures based on lexical, PoS, and syntactic patterns to analyze translationese in a dataset containing human translations with different levels of expertise, focusing on the automatic separation of non-translations from translations. This work addresses convergence as the proximity of two translation variants (professional and student). Register awareness is one of the critical elements of translation expertise (Olohan, 2015) — for example, the mentioned study by Redelinghuys (2016) show inexperienced translators to be more repetitive when translating creative writing than popular texts, which points to their practice in the academic context of translator training. A recent study by Popović (2020) explores differences between texts translated by professional translators, crowd contributors, and translation students, showing their impact on machine translation evaluation. This study suggests that it is crucial for machine translation evaluation to understand the factors influencing human translation variation, especially when we compare human and machine translation quality.

2.3 Register in translation

Our definition of register relies on variational linguistics (Biber, 1995; Halliday, 1985). Variation across registers is linked to the distribution of linguistic patterns in different contexts: register diversification represents distinctive distributions of linguistic patterns, as compared to the use of these patterns in other contexts (Biber et al., 1998, 13). Register variation has also been an object of analysis in translations. Kruger and van Rooy (2012) state that translationese is subject to the influence of register and Neumann (2013) demonstrates the degree to which translations get adapted to the requirements of different registers in English and German. Her feature set inspired the study by Evert and Neumann (2017) who detect similarities between register and translationese features. Lapshinova-Koltunski (2017) analyses the

interaction between register and translation method (human vs. machine), also paying attention to the differences between professional and novice translators. Lapshinova-Koltunski and Zampieri (2018) automatically discriminate registers and translation methods using part of speech n-grams. They show that it is harder to automatically differentiate between translation methods than between registers. This means that register diversification prevails over translation method diversification. This also points to a convergence between translations, which is of interest in our work. However, this convergence is related to the translations produced with different methods and not to the reduced distinction of various registers in favor of a more neutral “middle” register, as defined by Kruger and van Rooy (2012) and pursued in our work.

3 Research Design

3.1 Data

We use a dataset of English and German texts exported from two corpora. We derived English originals (EO), their translations into German by professionals (PT), as well as comparable German non-translations (GO) from the CroCo corpus (Hansen-Schirra et al., 2012). The non-professional translations (ST) for the same English sources as in CroCo were produced by students of translation and come from the corpus VARTRA (Lapshinova-Koltunski, 2013)². In this way, both professionals’ and students’ translations have the same sources and represent translation variants of the same original texts. Our dataset covers seven registers: political essays (ESSAY), fiction (FICTION), manuals (INSTR), popular-scientific articles (POPSCI), letters to shareholders (SHARE), prepared political speeches (SPEECH), and tourism leaflets (TOU). The English sources and the comparable German non-translated texts used for training our language models cover the same registers. In Table 1, we provide details on the size of the data under analysis.

To ensure the comparability of the models’ results in the source and the target languages, we use the Universal PoS tagset (Petrov et al., 2012). All texts in the data were automatically tokenized, lemmatized, and annotated with part of speech infor-

²We define professional translators as experts who have a good degree of experience in translating, mostly specializing in their areas, whereas students are trainees who have no or little experience in translation. While the two groups inhabit a continuum, we are happy with a binary division

	EO	GO	ST	PT
ESSAY	35 238	36 162	16 295	35 865
FICTION	37 019	36 913	12 755	37 953
INSTR	35 668	36 562	20 816	35 342
POPSCI	35 668	36 321	23 369	33 880
SHARE	36 437	35 517	25 630	36 810
SPEECH	35 223	35 769	24 999	36 377
TOU	35 981	36 564	20 358	34 139
TOTAL	251 894	253 862	144 222	250 366

Table 1: Dataset size in tokens.

mation based on the Universal Dependency framework (Straka and Straková, 2017). The accuracy of automatic annotation of the respective models for universal parts of speech is 90.5% for German and 94.5% for English³. Naturally, our PoS taggers can make mistakes, and it is conceivable that this margin of error might bring them to label some unusual sequences of words with more conventional, albeit erroneous, tags. Even if this anomaly were to happen, we find that it could not account for the differences we observe among our corpora since it would affect all texts similarly, and it would at worst slightly reduce their differences rather than magnify them.

While the amount of data is small for neural network training, it is essential to remember that since we are using a universal tagset, its vocabulary size is tiny: 15 parts of speech in total. This vocabulary size keeps the complexity of the learning process drastically lower than that of word sequence modeling and it allows our network to model small data well enough to display systematically lower perplexities when presented with unseen documents from the corpus on which it was trained (see for example Table 2).

3.2 Perplexity

We train two standard one-layered LSTM language models (LM) of 50 cells⁴ on the PoS sequences of 80% of the whole English and German non-translations respectively and measure their perplexity on professional translations, student translations, and originals. Even with small training data, our language models display lower perplexities for unseen instances of the originals from which we sampled the training set (see Table 2 in Section 4 below)

³See http://ufal.mff.cuni.cz/udpipe/models#universal_dependencies_20_models for details.

⁴We used Keras 2.0.9 (Chollet et al., 2015) running on Tensorflow 1.10.0 (Abadi et al., 2016)

than for translations in the same language.

We try two training sets: in the first case, we train LMs on the unweighted, randomly sampled 80% of the corpus. In the second case, we train our language models on a representative sample that respects the whole corpus’ genre percentages. In this way, we try to prevent domain bias from distorting our results in the test phase. In both cases, we test our models on unseen PoS sequences from originals or translations and analyze their average perplexity – a measure of how well a probability distribution predicts a sample as defined in (1), where $\{w_1, \dots, w_T\}$ is held out test data that provides the empirical distribution $q(\cdot)$ in the cross-entropy formula given in (2) and $p(\cdot)$ is the language model (LM) estimated on a training set.

$$PP = 2^{\tilde{H}_r} \quad \text{where} \quad \tilde{H}_r = -\frac{1}{T} \log_2 p(w_1, \dots, w_T) \quad (1)$$

$$\tilde{H} = -\sum_x q(x) \log p(x) \quad (2)$$

In this way, perplexity delivers a measure similar to surprisal in Information Theory (Shannon, 1948), according to which language items with high surprisal/ low predictability convey more information than items with low surprisal/ high predictability in context. Our analyses use neural language models’ average perplexity for the PoS n-grams in all the subcorpora under analysis. In terms of n-gram language models, predictability in context means $p(\text{unit}|\text{context})$, where context is the preceding context of n-1 words. A higher value of perplexity for a text means high surprisal/low predictability and, hence, an order of PoS sequences unusual for a reference corpus. We run perplexity-based tests for the remaining 20% of the non-translations and on both student and professional translations. We expect that the *relative* perplexity of English-trained and German-trained models (independently from their baselines) can tell us something about grammatical translationese.

We expect low perplexity values on the monolingual data (e.g. German non-translations on German non-translations) and high perplexity values on cross-lingual data (e.g. German non-translations on the English model). We also expect translations to fall between the source and the target language. In this way, perplexity values for translated data should be higher than those of non-translations within one language but lower than

the cross-lingual values. We also expect perplexity values for the professional translations to be lower than for the student translations within one language, but higher when tested cross-lingually (Hypothesis 1).

In terms of register diversification in the translated data, the essential idea is that an LM trained on a diverse set of registers⁵ will find, on average, a converging translation less perplexing, since it contains grammatical structures typical of what we could call “general language”. Thus we expect higher perplexity values for registers characterized by a distinctive or creative use of language – i.e., fiction – and lower values for more conventionalized registers – such as instruction manuals. Convergence will result in the homogeneity of perplexity values across different registers. Here, we expect a higher homogeneity, and hence convergence, for students than professionals (Hypothesis 2).

3.3 Pattern analysis

In the last step, we compare our perplexity results with the distributions of PoS n-grams across registers and corpora. Distributions of different PoS n-grams should show whether professional and student translators tend to be more repetitive or more diverse in using typical structures while translating various target language registers. So, we run a comprehensive examination of how many distinct PoS patterns translators use in a given text portion. Since our data contains the same source texts (and thus the same source patterns), we can expect that the more perplexing specific translations are, the more diverse patterns they should be.

We analyze PoS pattern diversity – the number of different PoS n-grams used in each register by students and professionals, which shows how many different PoS patterns translators use in a given portion of text and determine whether professionals are more diverse in translating registers than students. If students have an accentuated tendency to converge, they should show less diversity than professionals, which is especially revealing given that both professionals and students are translating the same source text, starting from the same source-structures. For all analyses, we have studied the differences between our subcorpora with growing n-grams, moving from bigrams up to heptagrams. We ran them on the same amount of text for all

⁵we trained LMs on the texts of the target language corpus that represent all registers.

subsets, thus down-sampling the professional and the original corpora.

4 Perplexity Score Analyses

4.1 Hypothesis 1

We report the perplexity scores not controlling for register in Table 2, which illustrates the results of the model performance on all the four subcorpora under analysis, as well as the results of the t-test showing that the models’ differences in perplexity are all statistically significant.

	EO-LM	GO-LM	t-value	p-value
EO	8.88	15.08	-11.6	<0.001
GO	11.12	5.93	23.5	<0.001
ST	12.51	11.12	3.2	0.001
PT	11.36	14.39	-10.1	<0.001

Table 2: Perplexity of the English-trained (EO-LM) and the German-trained models (GO-LM) on EO, GO, ST, and PT along with the results of t-test (t and p-value).

As stated in Section 3.2, we expect lower perplexity values for the tests within the monolingual data samples than for the cross-lingual data samples. Our English model is more surprised seeing other English PoS n-grams than German seeing other German PoS n-grams (8.88 vs. 5.93), which might derive from a more significant variation of morpho-syntactic patterns in the English data. A sanity check on the n-gram distribution shows that in our data, English has more diversity than German in terms of language patterns: for the vast majority of n-grams selections, English appears to have a higher number of different structures than German, which could be justified by the analytical character of English if compared to German: English uses more prepositions and auxiliaries to build up various constructions, whereas German expresses the same meaning through morphological strategies (endings, suffixes) that are not captured by the PoS n-grams. It is interesting to see that English and German are not equally surprised by each other: the English model is less surprised to see German n-grams (11.12) than is the German model when seeing English n-grams (15.08)⁶. Taking the language status of English, this might be, on the one hand, surprising as English has much influence on the German language, which takes over

⁶The differences between these distributions are statistically significant.

English structures (structural anglicisms). On the other hand, we can explain this difference again by the diversity of language patterns in the English data: we can expect a system modelled on English to be more used to structural change and, as such, less surprised by the new structures it encounters in German.

The English model is less perplexed by professional translations (11.36) than by non-professional ones (12.51). In this way, professionals seem to be closer to their source texts (interference). Student translations elicit a higher perplexity score (12.51), which indicates that they are even more surprising for the English model than the comparable German non-translations and translations by professionals, which indicates over-normalization – exaggerating the target language patterns as defined in Section 2.1. The German model’s results reveal an opposite tendency: professional translations seem to be more perplexing to the German model than the student ones. The German model seems to be highly surprised by the PoS sequences in the professional translations. Interpreting this result in terms of translationese, such a high level of perplexity, not far from the perplexity reached by English data, could indicate a degree of interference in professionals. This tendency is against our expectations formulated in Hypothesis 1 in Section 1.

4.2 Hypothesis 2

In the next step, we look into perplexity scores controlling for register in order to analyze convergence. We summarize our results in Table 3. We used a mixed training set that included a balanced number of sentences from each domain to maximize the data’s representativity.

	ST	PT	t-test	p-value
FICTION	11.41	12.74	-5.6	<.001
ESSAY	10.54	13.73	-14.2	<.001
POPSCI	10.20	10.50	-1.6	<.001
INSTR	8.59	9.63	-5.2	<.001
SHARE	12.65	13.23	-0.5	0.5
SPEECH	10.08	9.83	1.2	0.2
TOU	10.22	12.34	-9.04	.001
ALL	11.12	14.39	-2.45	0.01

Table 3: Perplexity of the German-trained model on ST and PT. We also report t-test and p value for each pair of distributions. We bolded the statistics that reject H_0 at the 0.05 significance level.

As seen from the table, all registers translated by professionals elicit higher scores than those translated by students, except for political speeches. However, the scores for this register, as well as those for letters to shareholders do not show a statistically significant difference between the two groups of translators. We interpret the lower scores of student translations as a reduced register distinction in favor of a more general language, which confirms our hypothesis that students are more repetitive in the language constructions they use. One of the reasons for this tendency could be that students tend to employ specific transfer patterns when translating from English into German, resulting in the frequent use of conventional structures and, consequently, a higher convergence of their translations. Another explanation could be that novice translators do not have enough knowledge about specific registers and various aspects of technical communication. Therefore, they translate different registers similarly, making them closer to the general language in German. Because they tend to repeat the same patterns for different registers, students seem less perplexing than professionals. We verify these assumptions in the experiments on pattern diversity in Section 5. We also compare the perplexity values across registers for both translation varieties. Using the scores in Table 3, we rank the registers for student and professional translations in Table 4.

We observe a very similar ranking for all registers in both translation variants. The only exception seems ESSAY, which is more distinct in professional translations and more conventionalized in student translations; one reason for this might be, as we will detail later, the over-normalization of other domains (i.e., FICTION) in student translations. The most conventionalized register in both translation varieties is INSTR. It is also interesting to see that the scores for registers translated by students are less variable than those for registers translated by professionals, which indicates register-related convergence in student translation.

5 Analysis of Pattern Diversity

Figure 1 illustrates the number of unique PoS n-grams used in the different registers of our German corpora by professionals (left graph) or students (right graph) – on the x-axis – as compared to the number of unique PoS n-grams used in the same registers by comparable German originals – on the y-axis.

PT	INSTR ⇒ SPEECH ⇒ POPSCI ⇒ TOU ⇒ FICTION ⇒ SHARE ⇒ ESSAY
ST	INSTR ⇒ SPEECH ⇒ POPSCI ⇒ TOU ⇒ ESSAY ⇒ FICTION ⇒ SHARE

Table 4: Register ranking according to their perplexity scores.

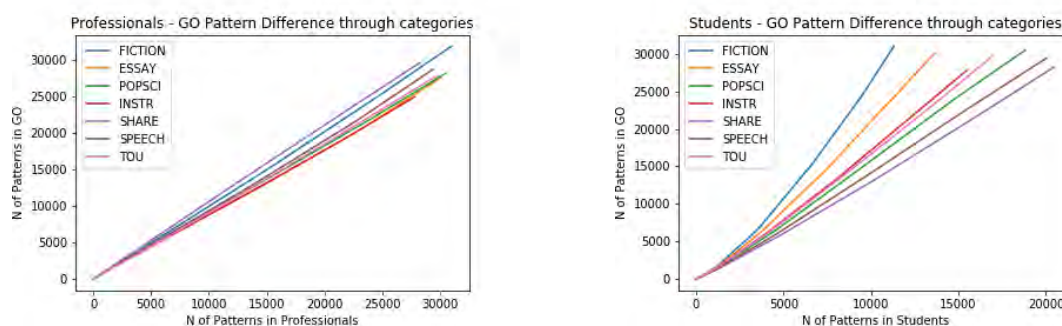


Figure 1: Differences between PoS n-grams in GO and PT (left side) and in GO and ST (right side), going from bigrams to heptagrams. For example, FICTION in GO has more than 30.000 different heptagrams, while FICTION in Students has about 10.000; instead, SPEECH progresses similarly for both categories through all ngrams, drawing a straighter line

We see from these graphs that professionals tend to have register-specific variations that are substantially similar to those of the equivalent originals, while students appear to be less diverse than both comparable originals and professionals, especially in more “creative” registers such as FICTION or ESSAY. Interestingly, the differences between professional and student translators (Figure 2) appear to be similar to those observed for ST and GO.

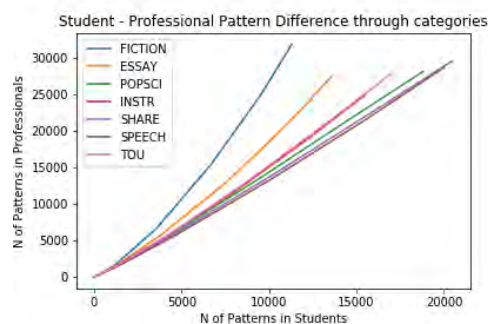


Figure 2: Differences between PoS patterns in the PT and the ST registers, going from bigrams to heptagrams, with heptagrams marking the end of each line.

It seems, overall, that the reason for the lower perplexity scores of the PoS-based language models for student translations is that students overnormalize their outputs, reusing fewer but more predictable structures. Professionals are more creative in their sentence structures: they are thus more perplexing for a general German model, but

their behavior is, paradoxically, more similar to that of original writers. We illustrate the differences in language patterns discovered between student and professional translators with examples (1) and (2). For this, we pick exemplars for which students turn to be more repetitive than professionals while translating the same text. We illustrate the pattern NOUN-ADP-DET-NOUN-DET-NOUN-ADP in student translation in (1), whereas the ST version in (2) displays an example of the VERB-ADP-DET-NOUN-ADJ-PUNCT-SCONJ structure.

- (1) a. *Seine Initialen, SR <...> waren in den Torbögen eingraviert und zogen sich durch das [Gebäude wie die Graffiti-malereien der Gangs in] den Straßen der Stadt.*
- b. *Und hier und da seine Initialen, SR <...> in Torbögen eingeritzt, [wie die Bandengraffiti] draussen auf der Strasse*
- c. *And his initials here and there, SR <...> carved in archways [like the gang graffiti in] the streets outside.*

In (1-a), the student translator uses a complex nominal structure and adds some information not available in the source. The translation by a professional in (1-b) contains the same information as in the source (1-c) and a more lexically dense structure (*Bandengraffiti* vs. *Graffitimalereien der Gangs*).

- (2) a. *Der Schweiß [lief an unserem Körper*

- herunter, sodass] unsere T-Shirts an unsere Rücken klebten.*
- b. *Der Schweiß [lief uns am Körper runter, daß] uns die Hemden am Rücken klebten.*
 - c. *The sweat [came down our bodies and] plastered our shirts to our backs.*

Both translation varieties in example (2) convey the same information from the source sentence. However, the translation by a professional in (2-b) sounds more natural in German, whereas the student translation in (2-a) is closer to the source sentence. The direct object in the English in (2-c) cannot be directly transferred into German because of the restriction on semantic diversity of subjects and objects in German. The professional translator changes the direct object into a Dative+prepositional object, whereas the student uses just a prepositional object.

6 Conclusion and Outlook

In this study, we analyzed translationese in professional and student translations using a perplexity-based approach. We modelled the source and target grammatical patterns with an LSTM architecture and tested the conformity to the source and target language conventions of the two translation varieties through PoS perplexity. Despite the relative scarcity of our data, the small vocabulary of universal PoS allowed our LSTMs to learn the short and long-distance patterns well enough to display significantly higher perplexities when confronted with translations instead of original texts. Through this method, we found that, surprisingly, professional translators elicit higher perplexity scores from the target language model than students, which is against our first hypothesis. Nonetheless, in the analysis of convergence, we tested the extent to which professional and student translations of various registers conform to the target language model. We found more convergence in student than in professional translations, confirming our second hypothesis. We then tried to understand such results by analyzing PoS n-gram patterns in both translation varieties and conducting a qualitative analysis of translation divergence in the data. Overall, we found that such higher perplexities are an artifact of higher register variation in professional translations. We are not observing interference, but rather professionals' essential ability to be more daring with their language use. Student translators converge

more, displaying a lower register sensitivity and a tendency to overuse the most general structures of the target language, while professional translators display more diversity and creativity in their structures, behaving in this way more similar to native writers.

The qualitative analysis of the examples suggest that the source of this diversity may originate from the cross-lingual differences between the source and the target languages: faced with a construction that has no direct or obvious equivalent in the target language, students might tend to choose less brilliant, more standard constructions across registers, whereas professionals might attempt to recreate the original domain's diversity. At the same time, we realize that our analyses may have some limitations. For instance, due to the absence of metadata in professional translations, we fail to control for individual variation in the data. For students, we know that the texts of various registers were sometimes translated by the same translators.

The results of our analyses provide an empirical contribution to the understanding of human translation. They show evidence of variation between texts generated by different translators in terms of language patterns and shed more light on the phenomenon of translationese. Studying variation in human translations of the same source texts across various registers is valuable for translation studies and multilingual language processing, especially for MT. As shown in Popović (2020), human translation variation plays a great role in MT evaluation.

In the future, we plan to deepen our understanding of how students over-normalize by aligning source and target texts, allowing for qualitative analyses of their translating behavior. We also want to explore whether other factors beyond the level of expertise influence translation convergence. Moreover, we would like to connect these results with the growing field of automatic translation quality estimation. Finally, although it is hard to find appropriate datasets containing comparable texts in terms of registers and different degrees of expertise, it would be interesting to expand this work on the opposite translation direction (German-English) and other language pairs to see if such tendencies are universally valid. Multilinguality would introduce more variance, and thus more factors to consider to avoid the risk of overclaiming and misunderstanding the complex phenomenon of translationese.

Acknowledgments

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Project-ID 232722074–SFB 1102.

References

- Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. TensorFlow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283.
- Mona Baker. 1996. Corpus-based translation studies: The challenges that lie ahead. In H.L. Somers, editor, *LSP and Translation: Studies in Language Engineering in Honour of Juan C. Sager*, page 175–186. John Benjamins Publishing Company, Amsterdam.
- Marco Baroni and Silvia Bernardini. 2006. A new approach to the study of translationese: Machine-learning the difference between original and translated text. *Literary and Linguistic Computing*, 21(3):259–274.
- Douglas Biber. 1995. *Dimensions of Register Variation: A Cross-Linguistic Comparison*. Cambridge University Press.
- Douglas Biber, Susan Conrad, and Randi Reppen. 1998. *Corpus Linguistics. Investigating Language Structure and Use*. Cambridge University Press, Cambridge.
- Yuri Bizzoni, Tom S Juzek, Cristina España-Bonet, Koel Dutta Chowdhury, Josef van Genabith, and Elke Teich. 2020. How human is machine translationese? Comparing human and machine translations of text and speech. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 280–290.
- Yuri Bizzoni and Elke Teich. 2019. Analyzing variation in translation through neural semantic spaces. In *Proceedings of the 12th Workshop on Building and Using Comparable Corpora (BUCC) at RANLP-2019*.
- François Chollet et al. 2015. Keras. <https://keras.io>.
- Gloria Corpas Pastor, Ruslan Mitkov, Naveed Afzal, and Lisette Garcia-Moya. 2008. Translation universals: do they exist? a corpus-based and nlp approach to convergence. In *Proceedings of the LREC-2008 Workshop on Building and Using Comparable Corpora*, pages 1–7.
- Stefan Evert and Stella Neumann. 2017. The impact of translation direction on characteristics of translated texts: A multivariate analysis for English and German. *Empirical Translation Studies: New Methodological and Theoretical Traditions*, 300:47.
- Martin Gellerstam. 1986. Translationese in Swedish novels translated from English. In L. Wollin and H. Lindquist, editors, *Translation Studies in Scandinavia*, pages 88–95. CWK Gleerup, Lund.
- Yvette Graham, Barry Haddow, and Philipp Koehn. 2020. Statistical power and translationese in machine translation evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 72–81, Online. Association for Computational Linguistics.
- Michael Alexander Kirkwood Halliday. 1985. *Spoken and Written Language*. Deakin University, Victoria.
- Silvia Hansen-Schirra, Stella Neumann, and Erich Steiner. 2012. *Cross-linguistic Corpora for the Study of Translations. Insights from the Language Pair English-German*. de Gruyter, Berlin, New York.
- Iustina Ilisei. 2012. *A machine learning approach to the identification of translational language: an inquiry into translationese*. Doctoral thesis, University of Wolverhampton.
- Iustina Ilisei, Diana Inkpen, Gloria Corpas Pastor, and Ruslan Mitkov. 2010. Identification of translationese: a supervised learning approach. In *Proceedings of CICLing-2010*, volume 6008 of *LNCS*, pages 503–511, Springer, Heidelberg.
- Haidee Kruger and Bertus van Rooy. 2012. Register and the Features of Translated Language. *Across Languages and Cultures*, 13(1):33–65.
- Maria Kunilovskaya and Ekaterina Lapshinova-Koltunski. 2020. Lexicogrammatic translationese across two targets and competence levels. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4102–4112, Marseille, France. European Language Resources Association.
- Veronika Laippala, Jenna Kanerva, Anna Missil, Anna Missilä, Sampo Pyysalo, Tapio Salakoski, and Filip Ginter. 2015. Towards the Classification of the Finnish Internet Parsebank : Detecting Translations and Informality. In *Nodalida*. Linköping University Electronic Press, Sweden.
- Ekaterina Lapshinova-Koltunski. 2013. VARTRA: A Comparable Corpus for Analysis of Translation Variation. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, pages 77–86, Sofia, Bulgaria. Association for Computational Linguistics.
- Ekaterina Lapshinova-Koltunski. 2017. Exploratory analysis of dimensions influencing variation in translation: The case of text register and translation method. In Gert De Sutter, Marie-Aude Lefer,

- and Isabelle Delaere, editors, *Empirical Translation Studies: New Methodological and Theoretical Traditions*, volume 300 of *TILSM series*, pages 207–234. Mouton de Gruyter. TILSM series.
- Ekaterina Lapshinova-Koltunski. 2020. Tracing normalisation and shining through in novice and professional translations with data mining techniques. In Sylviane Granger and Marie-Aude Lefer, editors, *Translating and Comparing Languages: Corpus-based Insights*, volume 6 of *Corpora and Language in Use Proceedings*, pages 33–47. Presses universitaires de Louvain, Louvain-la-Neuve.
- Ekaterina Lapshinova-Koltunski and Marcos Zampieri. 2018. Linguistic features of genre and method variation in translation: A computational perspective. In Th. Charnois, M. Larjavaara, and D. Legallois, editors, *The Grammar of Genres and Styles: From Discrete to Non-Discrete Units*, volume 320 of *TILSM series*, pages 92–117. Mouton de Gruyter.
- Sara Laviosa. 2002. *Corpus-based Translation Studies, Theory, Findings, Application*. Rodopi, Amsterdam.
- José Manuel Martínez Martínez and Elke Teich. 2017. Modeling routine in translation with entropy and surprisal: A comparison of learner and professional translations. In Larissa Cercel, Marco Agnetta, and Maria Teresa Amido Lozano, editors, *Kreativität und Hermeneutik in der Translation*. Narr Francke Attempto Verlag.
- Stella Neumann. 2013. *Contrastive register variation. A quantitative approach to the comparison of English and German*. Mouton de Gruyter, Berlin, Boston.
- Maeve Olohan. 2015. *Scientific and technical translation*. Routledge.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096.
- Maja Popović. 2020. On the differences between human translations. In *Proceedings of EAMT-2020*, Lisboa, Portugal.
- Ella Rabinovich, Noam Ordan, and Shuly Wintner. 2017. Found in Translation: Reconstructing Phylogenetic Language Trees from Translations. *Acl-2017*, pages 530–540.
- Karien Redelinghuys. 2016. Levelling-out and register variation in the translations of experienced and inexperienced translators: a corpus-based study. *Stellenbosch Papers in Linguistics*, 45(0):189–220.
- Raphael Rubino, Ekaterina Lapshinova-Koltunski, and Josef van Genabith. 2016. Information density and quality estimation features as translationese indicators for human translation classification. In *Proceedings of NAACL HT 2006*, pages 960–970, San Diego, California.
- Claude E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.
- Milan Straka and Jana Straková. 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99.
- Elke Teich. 2003. *Cross-Linguistic Variation in System and Text. A Methodology for the Investigation of Translations and Comparable Texts*. Mouton de Gruyter, Berlin.
- Gideon Toury. 1995. *Descriptive Translation Studies - and Beyond*, benjamins edition. John Benjamins Publishing Company.
- Gideon Toury. 2004. Probabilistic explanations in translation studies: Welcome as they are, would they qualify as universals? In A. Mauraanen and P. Kujamäki, editors, *Translation Universals: Do They Exist?*, Benjamins translation library, pages 15–32. J. Benjamins Publishing Company.
- Vered Volansky, Noam Ordan, and Shuly Wintner. 2015. On the features of translationese. *Digital Scholarship in the Humanities*, 30(1):98–118.
- Mike Zhang and Antonio Toral. 2019. The effect of translationese in machine translation test sets. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 73–81, Florence, Italy. Association for Computational Linguistics.

Appendix A. Additional Figures

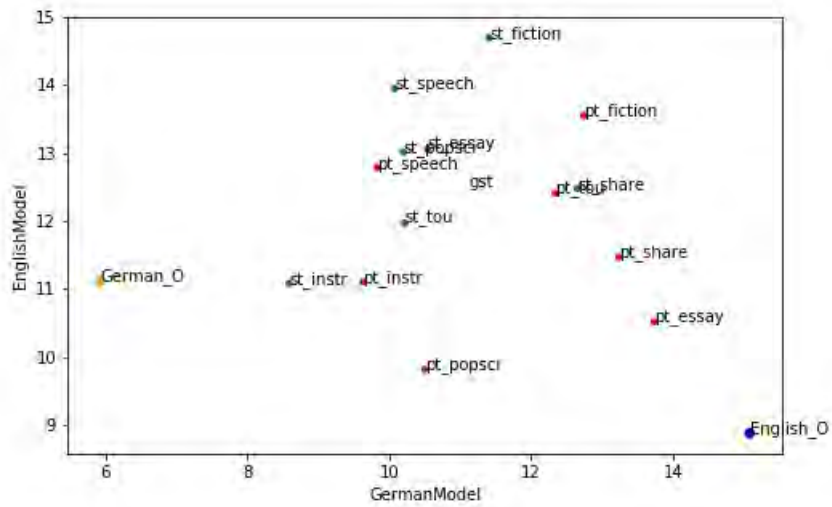


Figure 3: Perplexity of various registers of PT (red) and ST (green) for the English and German models, as well as general EO (blue) and general GO (yellow).

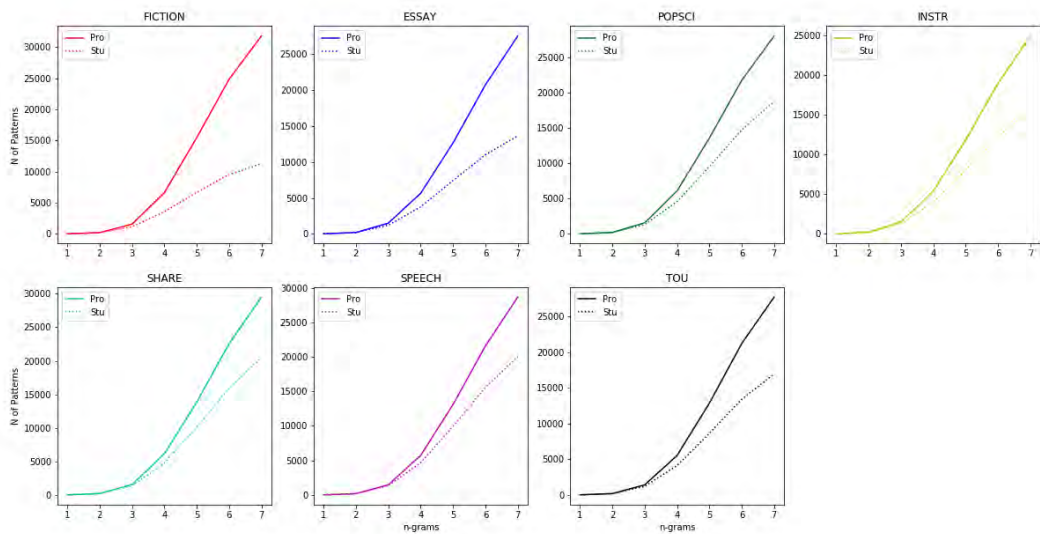


Figure 4: Number of different patterns used by students and professionals per each category, with growing n-gram length.

CombAlign: a Tool for Obtaining High-Quality Word Alignments

Steinþór Steingrímsson

Department of
Computer Science
Reykjavik University
Iceland
steinthor18@ru.is

Hrafn Loftsson

Department of
Computer Science
Reykjavik University
Iceland
hrafn@ru.is

Andy Way

ADAPT Centre
School of Computing
Dublin City University
Ireland
andy.way
@adaptcentre.ie

Abstract

Being able to generate accurate word alignments is useful for a variety of tasks. While statistical word aligners can work well, especially when parallel training data are plentiful, multilingual embedding models have recently been shown to give good results in unsupervised scenarios. We evaluate an ensemble method for word alignment on four language pairs and demonstrate that by combining multiple tools, taking advantage of their different approaches, substantial gains can be made. This holds for settings ranging from very low-resource to high-resource. Furthermore, we introduce a new gold alignment test set for Icelandic and a new easy-to-use tool for creating manual word alignments.

1 Introduction

Word alignment, the task of finding corresponding words in a bilingual sentence pair (see Figure 1), was a key component of statistical machine translation (SMT) systems. While word alignments are not necessary for neural machine translation (NMT), various MT methods incorporating word alignment have been found to achieve significant improvements in performance. Alkhouli et al. (2018) and Liu et al. (2016) use alignments as a

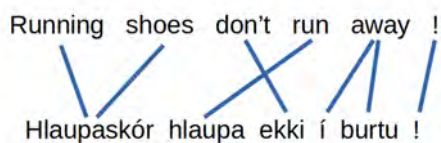


Figure 1: A simple example of English-Icelandic word alignments. Corresponding words are connected by edges.

prior; Arthur et al. (2016) augment NMT systems with discrete translation lexicons that encode low-frequency words; Press and Smith (2018) infer a correspondence between words in sentence pairs before encoding/decoding and, as demonstrated by Poncelas et al. (2019), back-translated data created using SMT systems, requiring word alignments, can be valuable to augment NMT systems. Word alignments have also been utilized to improve automatic post-editing (Pal et al., 2017) as well as to preserve markup in machine-translated texts (Müller, 2017).

Various other subfields of NLP make use of word alignments. Shi et al. (2021) show that by simply pipelining word alignment with unsupervised bitext mining, bilingual lexicon induction (BLI) quality can be improved significantly. For BLI, Artetxe et al. (2019) use an unsupervised MT pipeline, also employing word alignments. Kurfali and Östling (2019) use word alignments to filter noisy parallel corpora, and Paetzold et al. (2017) include word alignment as a part of their pipeline to align monolingual comparable documents.

There is a variety of word aligners available. *Giza++* (Och and Ney, 2003) and *fast_align* (Dyer et al., 2013) are easy to use implementations of the IBM models (Brown et al., 1993). Other statistical aligners, such as *eflomal* (Östling and Tiedemann, 2016), have also been shown to be fast and give competitive results. *SimAlign* (Masoud et al., 2020) takes advantage of the rising availability of contextualized embeddings and leverages them by extracting alignments from similarity matrices.

In this work, we present *CombAlign*, an ensemble of these four tools (*Giza++*, *fast_align*, *eflomal*, and *SimAlign*). As they are based on different approaches, and all able to attain a fairly high F_1 -score, it is reasonable to expect that combining their results in a sensible way could give better results than using any one of the individual systems.

Recently, the first reported results in SMT and NMT for Icelandic were published (Jónsson et al., 2020) within the context of an Icelandic national language technology programme (Nikulásdóttir et al., 2020). Icelandic is a morphologically rich West Germanic language with relatively few speakers, for which a substantial amount of language resources has been made available in recent years. However, no previous work has been conducted on word alignments for Icelandic. While testing our methods on four language pairs, we focus in particular on the effects of different alignment methods on the English-Icelandic (en-is) language pair. For finding the best hyperparameters for our ensemble, we thus do a grid search using an en-is development set.

Our main contribution is showing that it is possible to obtain high-quality word alignments using a combination of selected tools, outperforming all of the individual word alignment tools. We show this for four language pairs, with more detailed scrutiny of the results for one of them, en-is. Furthermore, we:

- publish a new gold standard word alignment reference set for en-is.
- make available a graphical tool, *AlignMan*, for manually curating word alignments.¹
- make the source code available for running the alignment tools and extracting combined alignments from them.²

2 Related Work

The most common statistical word alignment tools are based on the IBM models (Brown et al., 1993). These include *fast_align* (Dyer et al., 2013), *Giza++* (Och and Ney, 2003) and *eflomal* (Östling and Tiedemann, 2016), all used in this work. The five IBM models use lexical translation probabilities and probability distributions with the different models adding or emphasizing different features to tackle weaknesses of the other models. While *fast_align* builds on IBM model 2, *Giza++* iterates on a number of the models in sequence, as well as using an HMM model. *eflomal* uses a Bayesian model with Markov Chain Monte Carlo inference on the IBM models.

Several studies on word alignments in relation to neural models have been published. Liu et al.

(2016) show that attention can be seen as a re-ordering model as well as an alignment model, and Ghader and Monz (2017) investigate the differences between attention and alignment. Zenkel et al. (2019) apply stochastic gradient descent to directly optimize the attention activations towards a given target word, resulting in accurate word alignments, and Garg et al. (2019) extract discrete alignments from the attention probabilities learnt during regular NMT training and leverage them to optimize towards translation and alignment objectives. Most of these systems require parallel data for training, but *SimAlign* (Masoud et al., 2020) takes advantage of the rising availability of contextualized embeddings and leverages them by extracting alignments from similarity matrices induced from the embeddings, with no need for any external data.

Ensemble methods are common in NLP and, in many cases, have been shown to give more accurate results than using just one single approach. They have been used, for example, for classifying patent applications (Benites et al., 2018), for spellchecking (Stefanescu et al., 2011), POS-tagging (Henrich et al., 2009) and sentiment analysis (Araque et al., 2017). For word alignments, Tufiş et al. (2006) have previously used a union of two different alignment approaches, each producing distinct alignments. One of their aligners was an implementation of the IBM models, and the other used translation lexicons and phrase boundaries to detect alignments. Their combined aligner outperformed both individual systems, and its results produced approximately 10% fewer errors than the better individual aligner.

3 Data

For evaluation, we use gold standard word alignments for four language pairs: Czech, German, French and Icelandic, all paired with English (en-cs, en-de, en-fr and en-is, respectively). For the methods trained on parallel data, *Giza++*, *fast_align* and *eflomal*, we use a subset of 512k sentences from Europarl (Koehn, 2005), except in the case of Icelandic as detailed in Section 3.1. Further information on the test sets is given in Table 1.

3.1 Icelandic Data

No gold standard word alignments have previously been made available for Icelandic. In order

¹<https://github.com/steinst/AlignMan>

²<https://github.com/steinst/CombAlign>

Lang. Pair	Gold Standard	Sent. Pairs	Edges
en-cs	Mareček (2008)	2,501	67,424
en-de	Europarl ³	508	10,534
en-fr	Och and Ney (2000)	447	17,438
en-is	<i>new</i>	384	5,517

Table 1: Gold standard alignments used for evaluation. The en-is gold standard contains further 220 sentence pairs that were used as a development set for grid search.

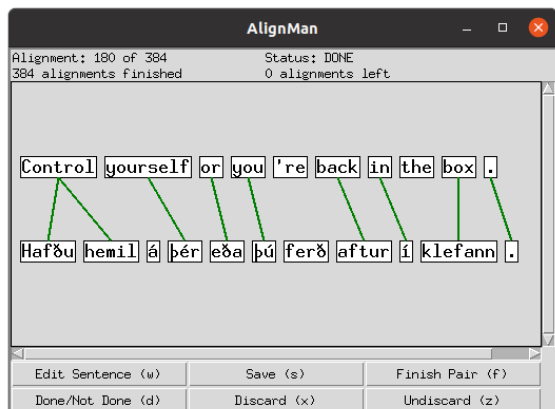


Figure 2: A screenshot from AlignMan. The program reads in text files with parallel sentences. The user can edit the sentences, discard them or create edges between words by moving the cursor to select corresponding words and then saving the alignment. It supports up to two users and can export a union or intersection of their alignments in two different formats.

to test our approach and other alignment methods on Icelandic, we thus compiled development and test sets. For that purpose, we created a simple graphical tool for performing manual word alignment, *AlignMan*, which is available under an Apache2 licence. A screen shot from AlignMan can be seen in Figure 2.

Two annotators manually aligned 604 sentences, a random sample from the *ParIce* en-is parallel corpus (Barkarson and Steingrímsson, 2019). They then reviewed the other annotator’s work in order to eliminate mistakes. The two annotations were then combined. All 1-to-1 alignments that

³<https://www-i6.informatik.rwth-aachen.de/goldAlignment/>

the annotators agreed upon were marked as ‘sure’ alignments and all other alignments made by either one or both of the annotators were marked as ‘possible’ alignments. The set was then split in two, with 220 sentences in a *dev*-set and 384 sentences in a *test*-set. The gold alignments are available for download from the CLARIN repository⁴ where further information on the criteria for building the corpus is available.

When parallel data was required to train the word aligners, sentence pairs from the ParIce corpus were used.

4 Methodology

In order to find the best settings for each aligner, we carry out a grid search. We run Giza++, fast_align and eflomal using different setups. For SimAlign, we use two different contextual embedding models and run them with different hyperparameters. We are thus working with five different aligners/alignment models. Finally, we proceed to find the best ensemble for different levels of parallel data availability.

4.1 Experimental Setup

By default, Giza++ runs IBM models 1, 3 and 4 as well as an HMM model, while fast_align is based on IBM model 2. We use default settings for these two aligners as well as for eflomal and compared their results after processing their output with different heuristics. These aligners are not trained on other word alignments, but rather on sentence-aligned parallel texts. They use an expectation maximization algorithm, iteratively learning from the parallel sentences; starting by initializing the model, then applying it to the data and setting the most probable alignments. After filling in gaps and collecting counts for particular word translations a new probability distribution is estimated. These steps are iterated until convergence.

Because the aligners learn probabilities from the data they run on, they should be better able to induce lexical translation probabilities and probability distributions when the size of the data increases, which in turn should lead to an increase in quality. In order to study this effect, we ran the aligners with varying numbers of sentences. The data we use for the experiments is described in Section 3.

⁴<http://hdl.handle.net/20.500.12537/103>

Giza++	
All settings default	
fast_align	
Heuristics	intersection , union, gd, gdf, gdfa
eflomal	
Heuristics	intersection , union, gd, gdf, gdfa
SimAlign	
Models	BERT, XLM-R
Tokenization	Word , BPE
Heuristics	Argmax , Itermax, Match
Distortion	[0.02, 0.03, ..., 0.09 , ..., 0.15]
Null extension	[0.85, 0.90, 0.95, 0.96, 0.97, 0.98, 0.99, 1.0]

Table 2: Hyperparameters for the different aligners. Shown in bold are the ones giving the highest F_1 -score.

Giza++ only outputs one set of alignments after each run, but for fast_align and eflomal we output alignments for both directions, source→target language and target→source, and then generate alignments from these using different alignment heuristics: intersection and union, as well as grow-diag (gd), grow-diag-final (gdf) and grow-diag-final-and (gdfa).

With SimAlign, we induce alignments from two different contextualized embedding models, multilingual BERT (mBert) (Devlin et al., 2019), and XLM-R (Conneau et al., 2020), and run experiments both for whole words and byte-pair encodings (BPE) (Sennrich et al., 2016). The alignments are obtained from similarity matrices using three different methods: *Match*, a graph-based method that identifies matches in a bipartite graph; *Argmax*, which aligns two words if the target word is the most similar to the source word, or vice versa; and *Itermax*, which applies Argmax iteratively and is thus better able to find alignment edges when one word aligns with two or more words in the other language. We did a grid search on the en-is development set, calculating the best scores using these methods and two other hyperparameters: distortion correction and null extensions, which set a threshold for when to remove edges and create null alignments. Different settings in our grid search are shown in Table 2.

For each of the alignment tools, we selected the hyperparameters giving the highest F_1 -score.

Then another grid search was carried out to find how best to combine the results. For that we had two parameters: combination of alignment tools, with 3 to 5 aligners/alignment models in each ensemble; and different parameters to join the alignments: with `unionall`, which accepts all alignments of the systems in the suggested ensemble, and different levels of intersection, from `intersectmin2` that requires two aligners to agree for an edge to be accepted, to `intersectmin5` where all aligners have to agree on each edge.

Finally, in order to examine whether our ensemble method is applicable to other language pairs, we test it on three of the test sets used in Masoud et al. (2020) and compare our results to theirs.

5 Experiments and Results

As described in Section 4.1, we identified the optimal settings and post-processing heuristics for each tool using grid search on the *dev*-set (see Table 2). We used these settings to obtain scores on our *test*-set, as shown in Tables 3 and 4.

5.1 Individual Aligners

While we use the same setting for each tool throughout, after having executed the grid search, the results of the ensemble differs in relation to how much data is being aligned. Relying at least in part on lexical translation probabilities, fast_align and Giza++ require a substantial amount of data before they become fairly accurate, while eflomal seems to be less susceptible to paucity of data. Figure 3 shows how F_1 increases for each system when evaluated on the Icelandic test set, when more parallel sentences are added for training. The aligners always learn from at least 384 test sentences, and up to an additional 3.6 million sentences. Table 3 shows precision, recall, F_1 -score and number of edges, i.e. individual word alignments, produced by eflomal, Giza++, and fast_align, when run with varying numbers of sentence pairs. Rather accurate from the start, the main advantage of training eflomal on more data is to get higher recall and more edges, while Giza++ and fast_align always output a similar number of edges, but both precision and recall rise when more sentence pairs are added.

SimAlign does not need any parallel data to learn from, and unlike the other aligners the results do not change when there is more data to

Samples	eflomal intersect				Giza++				fast_align intersect			
	Prec.	Rec.	F_1	Edges	Prec.	Rec.	F_1	Edges	Prec.	Rec.	F_1	Edges
0	.85	.76	.80	3803	.62	.74	.67	5387	.73	.67	.70	4005
1000	.87	.81	.84	4003	.64	.74	.68	5247	.78	.71	.74	3979
2000	.87	.83	.85	4098	.64	.75	.69	5223	.80	.73	.76	3978
4000	.87	.85	.86	4229	.64	.74	.68	5143	.82	.75	.78	3978
8000	.87	.87	.87	4320	.65	.74	.69	5117	.83	.76	.80	3976
16000	.88	.89	.88	4432	.67	.77	.72	5089	.85	.78	.81	3998
32000	.88	.90	.89	4507	.70	.79	.74	5072	.87	.80	.83	4008
64000	.88	.92	.9	4561	.72	.82	.77	5051	.88	.82	.85	4034
128000	.88	.93	.91	4622	.75	.85	.80	5019	.89	.84	.87	4086
256000	.88	.93	.91	4654	.78	.87	.82	5000	.90	.85	.88	4139
512000	.88	.93	.91	4667	.81	.89	.85	4982	.90	.86	.88	4151
1024000	.88	.94	.91	4713	.83	.91	.86	4951	.91	.87	.89	4165
2048000	.88	.94	.90	4722	.84	.91	.87	4927	.91	.86	.89	4139
3600000	.88	.94	.91	4745	.85	.92	.88	4913	.91	.86	.89	4115

Table 3: Precision, recall, F_1 -scores and number of edges for each of the IBM model-based aligners, with various numbers of parallel sentences added for training the aligners.

align. However, the tokenization used (BPE or the original word forms) and how the alignments are obtained from the similarity matrix, has a substantial effect on the resulting alignments, as seen in Table 4. The table shows that ArgMax gives a substantially higher precision than IterMax and Match, but since IterMax has higher recall, the F_1 -scores are quite close.

5.2 Ensembles

As can be seen in Table 3, eflomal does not need much training data to reach high precision. Thus, it should not be surprising that in low-resource scenarios a combination of eflomal with the two

unsupervised SimAlign models gives the best results. When more data is available, the other two IBM-model based aligners become more accurate, and as a consequence, more useful in an ensemble.

We thus report on two different ensembles: *EnsembleSmall*, comprised of three aligners which is better in cases where there is scarce data, and *EnsembleLarge* which uses all five aligners. Our ensemble strategy is simple: for both ensembles we only require a majority vote on each alignment. For EnsembleSmall we thus require 2 out of 3 aligners to suggest an alignment candidate for it to be accepted. EnsembleSmall uses the alignments produced by SimAlign’s *IterMax*, which

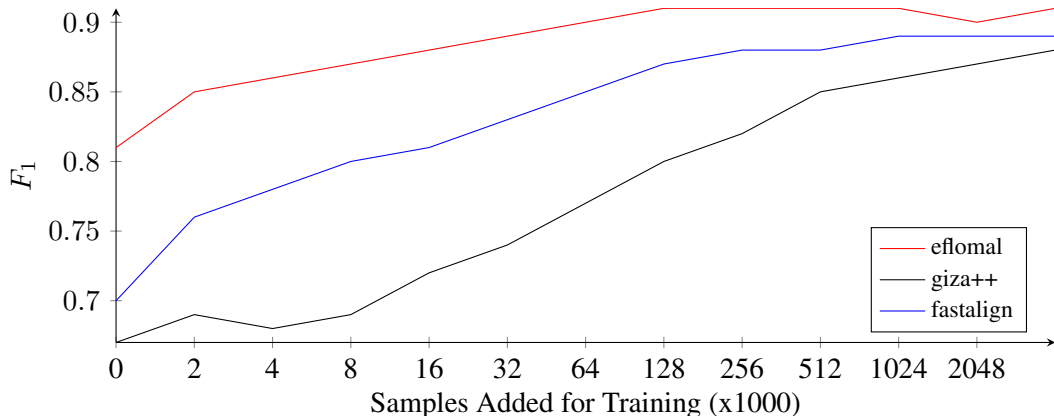


Figure 3: F_1 for word alignments generated using different alignment tools as a function of the number of sentence pairs used for training. F_1 for SimAlign-mBERT is 0.86 and 0.90 for SimAlign-XLM-R.

SimAlign						
Model	Tok.	H.	Pr.	Rc.	F_1	Edg.
mBERT	BPE	AM	.85	.84	.84	4468
		IM	.74	.91	.82	5717
		M	.66	.92	.77	6590
	word	AM	.88	.84	.86	4145
		IM	.79	.90	.84	5111
		M	.75	.91	.82	5463
XLM-R	BPE	AM	.88	.90	.89	4599
		IM	.78	.94	.86	5615
		M	.69	.96	.80	6618
	word	AM	.92	.88	.90	4165
		IM	.85	.93	.89	4925
		M	.78	.94	.86	5473

Table 4: Precision, F_1 -measure and number of edges for different setups of SimAlign. All these settings use 0.09 for distortion. The heuristics are: AM=ArgMax, IM=IterMax, M=Match.

has higher recall, an advantage when only one of the aligners in the ensemble is allowed to miss an alignment. EnsembleLarge requires 3 out of 5 aligners to agree and uses SimAlign’s *ArgMax*, which has more precision. Figure 4 shows how the F_1 -scores for the two ensembles rise with more data, and how EnsembleLarge, being more reliant on data, needs only tens of thousands of sentence pairs to outperform EnsembleSmall which obtains higher F_1 -scores in very low-resource settings. In contrast, EnsembleLarge, always having higher precision as shown in Table 5, produces fewer edges.

Our combination is based on a majority vote and the ensemble obtaining the highest F_1 -score is selected. Accordingly, it is possible to obtain higher precision using other combinations in situations where precision is critical and recall is not as important. This could be realised by setting a higher requirement for agreement between aligners, raising the precision even further, but at the price of retrieving fewer edges and thus a lower F_1 -score. For higher recall, lowering the agreement requirements works, although at the cost of some precision. Table 5 shows the combinations giving the best precision and F_1 -score, as well as recall and number of edges suggested by the system.

CombAlign					
Samples	Ensemble	Prec.	Rec.	F_1	Edges
0	EnsSm	.92	.92	.92	4410
	EnsLa	.93	.81	.87	3743
1000	EnsSm	.92	.93	.92	4458
	EnsLa	.94	.84	.89	3819
2000	EnsSm	.91	.93	.92	4459
	EnsLa	.95	.85	.90	3852
4000	EnsSm	.91	.93	.92	4506
	EnsLa	.95	.86	.90	3866
8000	EnsSm	.91	.94	.92	4529
	EnsLa	.95	.87	.91	3933
16000	EnsSm	.91	.94	.93	4569
	EnsLa	.96	.88	.92	3970
32000	EnsSm	.91	.95	.93	4591
	EnsLa	.96	.90	.93	4025
64000	EnsSm	.91	.95	.93	4624
	EnsLa	.96	.91	.93	4070
128000	EnsSm	.91	.95	.93	4635
	EnsLa	.96	.92	.94	4147
256000	EnsSm	.91	.95	.93	4656
	EnsLa	.96	.92	.94	4178
512000	EnsSm	.91	.95	.93	4648
	EnsLa	.96	.93	.94	4220
1024000	EnsSm	.91	.95	.93	4653
	EnsLa	.96	.94	.95	4249
2048000	EnsSm	.90	.95	.93	4679
	EnsLa	.96	.94	.95	4266
3600000	EnsSm	.90	.95	.93	4681
	EnsLa	.96	.94	.95	4265

Table 5: Precision, recall, F_1 -scores and number of edges for different setups of the CombAlign ensemble.

5.3 Utilizing the Word Alignments

As noted in Section 1, word alignments can be used for many different purposes, sometimes using SMT systems as intermediaries. In order to see whether our alignments are beneficial for SMT systems, we trained three Moses models, keeping all components of the training process the same, except for word alignments. For training, we used the data and filtering methods described in Jónsson et al. (2020).

Our baseline system uses the default Moses settings, with Giza++ for word alignments. We trained two other models, *CombAlignF1*: using the settings giving the highest F_1 -score as detailed in Section 5.2; and *CombAlignRec*: where we are still using the five aligners in the ensemble, but are more lenient and only require two or more of the five aligners to be in agreement. We did this as our highest scoring ensemble, *CombAlignF1*, generates 15% fewer edges than Giza++ and, for this task, recall is likely to be important. By relaxing the demands for agreement between the aligners, we raise recall while still only generating a similar number of edges between words as Giza++.

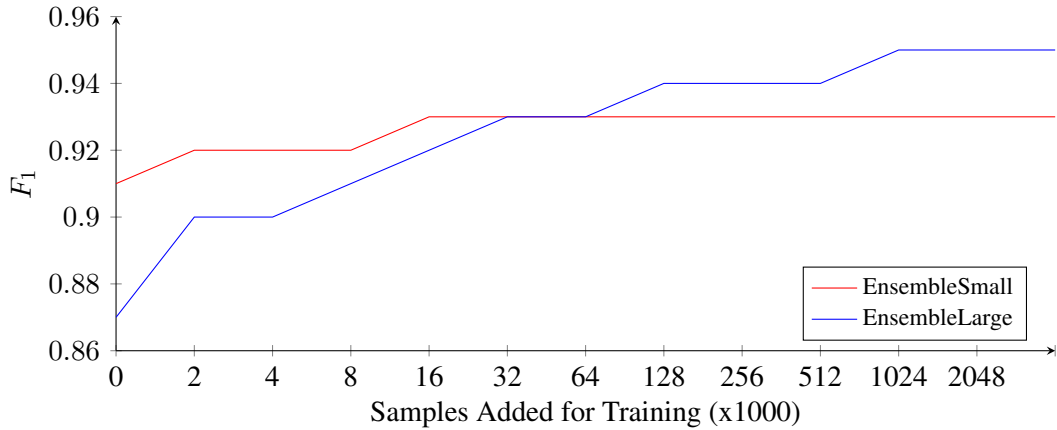


Figure 4: F_1 score for aligner ensembles. *EnsembleSmall* uses three alignment models and *EnsembleLarge* uses all five alignment models, as described in Section 5.2.

We compared these three systems in the following manner. First, we examined the phrase tables generated during training. The baseline system creates a phrase table with 3,496K lines, *CombAlignFI* has 1,319K lines and *CombAlignRec* has 1,774K lines. Manual inspection shows that the removed lines are almost always faulty so this pruning should not have negative effects on the system. Second, we tested the systems, using the three test sets from Jónsson et al. (2020), calculated the BLEU scores and manually inspected and evaluated the differences in translation.

BLEU scores for *CombAlignFI* were almost the same as for the baseline system, with a difference ranging from 0.01 to 0.11 for the three test sets. *CombAlignRec* had slightly better scores, scoring 0.4 to 0.95 higher BLEU than the baseline system.

We then manually compared a random sample of 450 translated sentences from the baseline system and *CombAlignRec*. 46% of the outputs were exactly the same; 14% had multiple faults for both systems and were deemed equally bad; 17% of the sentences were translated better by the baseline system and 23% had better translations produced by *CombAlignRec*. We categorized the errors made by the systems and while the sample size is quite small, and there is no clear distinction between the systems, *CombAlignRec* seems to be more likely to have errors when there are multiple numerical tokens in the sentence to translate, possibly because they may be treated like rare words. Moreover, *CombAlignRec* seems less likely to have words missing in the translated output and it seems more likely to make a more appropriate lexical choice, both in terms of content

words and verb inflections. A more thorough investigation is needed to understand why this is the case.

5.4 Other Language Pairs

In order to show that the ensemble methods work for other languages than Icelandic, we ran an experiment on three test sets. Table 6 shows the results and a comparison to the previous best, as reported on in Masoud et al. (2020).

In this experiment, we used two settings for the IBM-model based alignment tools: only running on the test-set data, and running with additional parallel data of 512K sentence pairs for training each language pair. Although the results for *CombAlign* always outperform the individual aligners, the difference is not always as large as for the en-is language pair. This may possibly be explained by the fact that the contextualized embeddings have more data on the other languages and thus give better predictions than when predicting Icelandic, or that the parallel training data is not in the same domain as the test sets, while the Icelandic test sets contained sentence pairs sampled from the parallel corpus (ParIce) used for training.

For the best-scoring ensembles, we used SimAlign’s *Itermax* when the statistical aligners used parallel data as well as when no additional data was used. This was due to *Itermax* giving the highest F_1 -score for these language pairs. This was not true for Icelandic, possibly because the contextual models were trained on less Icelandic data and so have more ‘knowledge’ of these other languages than it has of Icelandic.

Method	cs-en		en-fr		en-de	
Train. data (K)	0	512	0	512	0	512
eflomal	.79	.86	.82	.91	.61	.73
fast_align	.66	.78	.73	.86	.52	.70
Giza++	.71	.81	.69	.89	.55	.73
SimAlign:						
XLM-R	.87		.93		.78	
SimAlign:						
BERT	.87		.94		.81	
Previous work	.87		.94		.81	
CombAlign	.89	.91	.95	.95	.80	.83

Table 6: Word alignment F_1 -scores for cs-en, en-fr and en-de language pairs, with or without using training data.

6 Conclusion and future work

We have shown that using a very simple combination method for word alignment, it is possible to increase the accuracy substantially, both in low- and high-resource settings.

We evaluated on four language pairs, *en-cs*, *en-de*, *en-fr* and for the first time *en-is*, for which we manually created a new gold standard word alignment reference set. In order to do that we created and published a graphical tool for manual word alignments.

While our method uses minimal data processing, some pre-processing like POS-tagging and lemmatizing may raise the accuracy even further, especially in the case of a morphologically rich language like Icelandic. A comparison of typical misalignments per aligner is also likely to be beneficial, as knowing these properties might help in combining the aligners more effectively. The mBERT and XLM-R models we employ through SimAlign give good results, but there may still be room for improvement, for instance by pre-training these models on more Icelandic texts, which are scarce in the multilingual training corpus. It may also be worth considering to train a bilingual word embedding model and use that for alignment instead of, or in combination with, the other contextualized embedding models.

In the paper, we reported on preliminary results from training an SMT system using our word alignments. We plan to investigate whether the slightly better SMT output will be more beneficial for back-translations to augment NMT systems, following Poncelas et al. (2019). We also plan to compare BLI quality using the setup in (Artetxe

et al., 2019) and the same setup using our alignments. Furthermore, we intend to apply our alignments to training alignment-assisted NMT transformer models, by adding an alignment attention layer as described in (Alkhouli et al., 2018).

Acknowledgements

This work is supported by the Language Technology Programme for Icelandic 2019-2023, funded by the Icelandic government, and by the ADAPT Centre for Digital Content Technology which is funded under the Science Foundation Ireland (SFI) Research Centres Programme (Grant No. 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

- Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. On The Alignment Problem In Multi-Head Attention-Based Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 177–185, Brussels, Belgium.
- Oscar Araque, Ignacio Corcuera-Platas, J. Fernando Sánchez-Rada, and Carlos A. Iglesias. 2017. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77:236 – 246.
- Mikel Artetxe, Gorika Labaka, and Eneko Agirre. 2019. Bilingual Lexicon Induction through Unsupervised Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5002–5007, Florence, Italy.
- Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating Discrete Translation Lexicons into Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Austin, Texas.
- Starkaður Barkarson and Steinþór Steingrímsson. 2019. Compiling and Filtering ParIce: An English-Icelandic Parallel Corpus. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 140–145, Turku, Finland.
- Fernando Benites, Shervin Malmasi, and Marcos Zampieri. 2018. Classifying Patent Applications with Ensemble Methods. In *Proceedings of the Australasian Language Technology Association Workshop 2018*, pages 89–92, Dunedin, New Zealand.
- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota.
- Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia.
- Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly Learning to Align and Translate with Transformer Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China.
- Hamidreza Ghader and Christof Monz. 2017. What does Attention in Neural Machine Translation Pay Attention to? In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 30–39, Taipei, Taiwan.
- Verena Henrich, Timo Reuter, and Hrafn Loftsson. 2009. Combitagger: A system for developing combined taggers. In *Proceedings of the 22nd International FLAIRS Conference*, pages 254–259, Sanibel Island, Florida.
- Haukur Páll Jónsson, Haukur Barri Símonarson, Vésteinn Snæbjarnarson, Steinþór Steingrímsson, and Hrafn Loftsson. 2020. Experimenting with Different Machine Translation Models in Medium-Resource Settings. In *Proceedings of Text, Speech, and Dialogue – 23rd International Conference*, volume 12284 of *Lecture Notes in Computer Science*, pages 95–103.
- Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand.
- Murathan Kurfalı and Robert Östling. 2019. Noisy Parallel Corpus Filtering through Projected Word Embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 277–281, Florence, Italy.
- Lemao Liu, Masao Utiyama, Andrew Finch, and Ei-ichiro Sumita. 2016. Neural Machine Translation with Supervised Attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan.
- David Mareček. 2008. Automatic Alignment of Teletogrammatical Trees from Czech-English Parallel Corpus. Master’s thesis, Charles University.
- Jalili Sabet Masoud, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online.
- Mathias Müller. 2017. Treatment of Markup in Statistical Machine Translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 36–46, Copenhagen, Denmark.
- Anna Nikulásdóttir, Jón Guðnason, Anton Karl Ingason, Hrafn Loftsson, Eiríkur Rögnvaldsson, Einar Freyr Sigurðsson, and Steinþór Steingrímsson. 2020. Language Technology Programme for Icelandic 2019-2023. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3414–3422, Marseille, France.
- Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong.
- Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.
- Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.
- Gustavo Paetzold, Fernando Alva-Manchego, and Lucia Specia. 2017. MASSAlign: Alignment and Annotation of Comparable Documents. In *Proceedings of the IJCNLP 2017, System Demonstrations*, pages 1–4, Taipei, Taiwan.
- Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, Qun Liu, and Josef van Genabith. 2017. Neural Automatic Post-Editing Using Prior Alignment and Reranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 349–355, Valencia, Spain.
- Alberto Poncelas, Maja Popović, Dimitar Shterionov, Gideon Maillette de Buy Wenniger, and Andy Way. 2019. Combining PBSMT and NMT Back-translated Data for Efficient NMT. In *Proceedings*

of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), pages 922–931, Varna, Bulgaria.

Ofir Press and Noah A. Smith. 2018. You May Not Need Attention. *ArXiv*, abs/1810.13409.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Haoyue Shi, Luke Zettlemoyer, and Sida I. Wang. 2021. Bilingual Lexicon Induction via Unsupervised Bitext Construction and Word Alignment. *ArXiv*, abs/2101.00148.

Dan Stefanescu, Radu Ion, and Tiberiu Boros. 2011. TiradeAI: An Ensemble of Spellcheckers. In *Proceedings of the Spelling Alteration for Web Search Workshop*, pages 20–23, Bellevue, Washington.

Dan Tufiş, Radu Ion, Alexandru Ceauşu, and Dan Ştefănescu. 2006. Improved Lexical Alignment by Combining Multiple Reified Alignments. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 153–160, Trento, Italy.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding Interpretable Attention to Neural Translation Models Improves Word Alignment. *ArXiv*, abs/1901.11359.

Understanding Cross-Lingual Syntactic Transfer in Multilingual Recurrent Neural Networks

Prajit Dhar Arianna Bisazza

Center for Language and Cognition

University of Groningen

{p.dhar, a.bisazza}@rug.nl

Abstract

It is now established that modern neural language models can be successfully trained on multiple languages simultaneously without changes to the underlying architecture. But what kind of knowledge is really shared among languages within these models? Does multilingual training mostly lead to an alignment of the lexical representation spaces or does it also enable the sharing of purely grammatical knowledge? In this paper we dissect different forms of cross-lingual transfer and look for its most determining factors, using a variety of models and probing tasks. We find that exposing our LMs to a related language does not always increase grammatical knowledge in the target language, and that optimal conditions for *lexical-semantic* transfer may not be optimal for *syntactic* transfer.

1 Introduction

One of the most important NLP discoveries of the past few years has been that a *single* neural network can be successfully trained to perform a given NLP task in *multiple* languages without architectural changes compared to monolingual models (Östling and Tiedemann, 2017; Johnson et al., 2017). Besides important practical advantages (fewer parameters and models to maintain), such multilingual Neural Networks (mNNs) provide an easy but powerful way to transfer task-specific knowledge from high- to low-resource languages (Devlin et al., 2019; Conneau and Lample, 2019; Aharoni et al., 2019; Neubig and Hu, 2018; Arivazhagan et al., 2019; Artetxe and Schwenk, 2019; Chi et al., 2020). These success stories have led to a need for understanding *how* exactly cross-lingual transfer works within

these models. Figure 1 illustrates different possible characterizations of a trained mNN: While the no-transfer scenario is rather easy to rule out, understanding which linguistic categories are shared, and to what extent, is more challenging.

In this work, we focus on the transfer of *syntactic* knowledge among languages and look for evidence that mNNs induce a shared syntactic representation space *while not receiving any direct cross-lingual supervision*. To be clear, if we measure transfer among languages X and Y, every training sentence for language modeling will be either in language X or Y, while for machine translation every sentence pair will be either in language pair (X, Z) or (Y, Z). Thus, the only pressure to share linguistic representations is given by the sharing of the hidden layer parameters (as well as, possibly, some of the word embeddings).

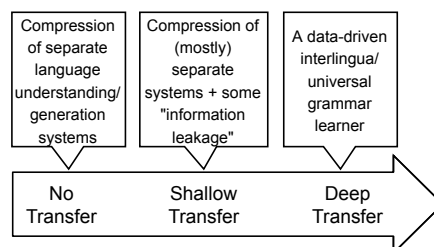


Figure 1: Possible characterizations of a trained mNN in terms of cross-lingual transfer levels.

Neural language models have been shown to implicitly capture non-trivial structure-sensitive phenomena like long-range number agreement (Linzen et al., 2016; Gulordava et al., 2018; Marvin and Linzen, 2018). However most of these studies have been confined to monolingual models. We then investigate the following questions:

1. Does mNNs' implicit syntactic knowledge of L2 increase by exposure to a related L1?
2. Do mNNs induce a common representation space with shared syntactic categories?

Our research questions are reminiscent of well-known questions in the fields of psycholinguistic and second language acquisition, where work has shown that both lexical and syntactic representations are shared in the mind of bilinguals (Hartsuiker et al., 2004a; Vasilyeva et al., 2010). Taking inspiration from this body of work, we investigate what factors are needed for mNNs to successfully transfer linguistic knowledge, including vocabulary overlap, language relatedness, number of training languages, training regime (joint *vs* sequential) and training objective (next word prediction *vs* translation to a third language).

In contrast to the current mainstream focus on BERT-like models (Rogers et al., 2020), we evaluate more classical LSTM-based models trained for next word prediction or translation over a moderate number of languages (2 or 9). We choose this setup because (i) it allows for more controlled and easy-to-replicate experiments in terms of both training data and model configuration and (ii) LSTMs trained on a standard sequence prediction objective are more cognitively plausible and directly applicable to our main probing task, namely agreement prediction. In this setup, we find limited and rather inconsistent evidence for the transfer of implicit grammatical knowledge when semantic cues are removed (Gulordava et al., 2018). While moderate PoS category transfer occurs, truly language-agnostic syntactic categories (such as *noun* or *subject*) do not seem to emerge in our mNN representations. Finally, we find that optimal conditions for lexical-semantic transfer may not be optimal for syntactic transfer.

2 Previous Work

Multilingual Machine Translation Early work on multilingual NMT focused on building dedicated architectures (Dong et al., 2015; Firat et al., 2016; Johnson et al., 2017). Starting from (Johnson et al., 2017), m-NMT models are mostly built with the same architecture as their monolingual counterparts, by simply adding language identifying tags to the training sentences. Using a small set of English sentences and their Japanese and Korean translations, Johnson et al. (2017) showed that semantically equivalent sentences form well-defined clusters in the high-dimensional space induced by a NMT encoder trained on large-scale proprietary datasets. Kudugunta et al. (2019) analyze the similarity of encoder representations of

different languages within a massively m-NMT model. They find that representation similarity correlates strongly with linguistic similarity and that encoder representations diverge based on the target language. However they do not disentangle the syntactic aspect from other types of transfer.

Multilingual Sentence Encoders A related line of work focuses on mapping sentences from different languages into a common representation space to be used as features in downstream tasks where training data is only available in a different language than the test language. Artetxe and Schwenk (2019) use the encoder representations produced by a massively multilingual NMT system similar to (Johnson et al., 2017) to perform cross-lingual textual entailment (XNLI) and document classification. m-BERT (Devlin et al., 2019; Devlin, 2018) and XLM (Conneau and Lample, 2019) are large-scale mNNs trained on a masked LM (MLM) objective using mixed-language corpora. This results in general-purpose contextualized word representations that are multilingual in nature, *without* requiring any parallel data. m-BERT representations have been proved particularly successful for transferring dependency parsers to low- (or zero-)resource languages (Wu and Dredze, 2019; Kondratyuk and Straka, 2019; Tran and Bisazza, 2019). On the task of cross-lingual textual entailment (Conneau et al., 2018b), XLM-based classifiers come surprisingly close to systems that use fully-supervised MT as part of their pipeline (to translate the training or test data).

Implicit Learning of Linguistic Structure NNs trained for downstream tasks such as language modeling, translation or textual entailment, have been shown to implicitly encode a great deal of linguistic structure such as morphological features (Belinkov et al., 2017; Bisazza and Tump, 2018; Bjerva and Augenstein, 2018), number agreement (Linzen et al., 2016; Gulordava et al., 2018) and other structure-sensitive phenomena (Marvin and Linzen, 2018). Studies such as (Tenney et al., 2019b,a; Jawahar et al., 2019) have extended these findings to BERT representations showing positive results on a variety of syntactic probing tasks. Extensive overviews of this body of work are presented in (Belinkov and Glass, 2019) and (Rogers et al., 2020).

Cross-lingual Transfer in Multilingual NNs Recent studies (Wu and Dredze, 2019; Pires et al.,

2019; Chi et al., 2020) have found evidence of *syntactic* transfer in m-BERT using POS tagging and dependency parsing experiments. On the other hand, Libovický et al. (2019) find that m-BERT representations capture cross-lingual *semantic* equivalence sufficiently well to allow for word-alignment and sentence retrieval, but fail at the more difficult task of MT quality estimation. While this massive Transformer-based (Vaswani et al., 2017) architecture has received overwhelming attention in the past year, we believe that smaller, better understood, and easier to replicate model configurations can still play an important role in the pursuit of NLP model explainability. Moreover, the large number of m-BERT training languages (ca. 100) added to the uneven language data distribution and the highly shared subword vocabulary, make it difficult to isolate transfer effects in any given language pair. Mueller et al. (2020) recently tested a LSTM trained on five languages on a multilingual extension of the subject-verb agreement set of Marvin and Linzen (2018). They found signs of harmful interference rather than positive transfer across languages. In Section 4 we corroborate this rather surprising finding by using a more favourable setup for transfer, that is: (i) only two, related, training languages, (ii) a simulated low-resource setup for the target language, and (iii) eliminating vocabulary overlap during training with language IDs.

Cross-lingual Transfer in the Bilingual Mind

Measuring the extent to which dual-language representations are shared in the mind of bilingual subjects is a long-standing problem in the field of second language acquisition (Kellerman and Sharwood Smith, 1986; Odlin, 1989; Jarvis and Pavlenko, 2008; Kootstra et al., 2012). Among others, Hartsuiker et al. (2004b) present evidence of cross-lingual *syntactic priming* in bilingual English-Spanish speakers, which are more inclined to produce English passive sentences after having heard a Spanish passive sentence. Using neuroimaging techniques in a reading comprehension experiment with in German-English bilinguals, Tooley and Traxler (2010) report that the processing of L1 and L2 sentences activates the same brain areas, pointing to the shared nature of syntactic processing in the bilingual mind. Taking inspiration from this body of work, we investigate what factors trigger cross-lingual transfer of syntactic knowledge within mNNs.

Cross-Lingual Dependency Parsing Finally, our work is also related to the productive field of cross-lingual and multilingual dependency parsing (Naseem et al., 2012; Zhang and Barzilay, 2015; Täckström et al., 2012; Ammar et al., 2016, *inter alia*), with the important difference that we are interested in models that are *not* explicitly trained to recognize syntactic structure but acquire it indirectly while optimizing next word prediction or translation objectives. Among others, Ahmad et al. (2019) have shown that the difficulty of transferring a dependency parser cross-lingually depends on typological differences between the source and target languages, with word order differences playing an important role. In this paper, we mainly consider source-target languages that are related, like French or Spanish (source) and Italian (target), where we expect implicit syntactic knowledge to be more easily transferable.

3 Probing Tasks

To answer our RQ1 (are mNNs capable of implicitly transferring syntactic knowledge between languages?) we choose the task of Number Agreement. For our RQ2 (are mNNs able to induce a common representation with shared syntactic categories?) we look at less complex syntactic tasks such as PoS tag classification and Dependency relation classification, and contrast them with a lexical-semantic task (word translation retrieval). We choose these tasks because they can be framed as simple classification (or ranking) problems and have a direct linguistic interpretation. We do not consider parsing because it is a complex task with a highly structured prediction space requiring dedicated model components. The probed models are LSTM-based language models and translation models, trained at the word-level. More details are provided below.

3.1 Number Agreement

Number agreement describes the instance where a phrase and its arguments or modifiers must agree in their number feature. Number agreement can occur between a subject-predicate pair (*the son_{sg} of my neighbors goes_{sg}*), noun-quantifier pair (*many_{pl} huge trees_{pl}*), etc. Linzen et al. (2016) first proposed the subject-verb agreement task to assess the ability of a LSTM-based LM to capture non-trivial language structure, by checking if the correct verb form was assigned a higher probabil-

ity than the wrong one, e.g. if $\text{prob}(\textit{were}|\textit{context}) > \text{prob}(\textit{was}|\textit{context})$ in the sentence *The boys, who were lost in the forest were/was found.* LM performance was shown to be mostly affected by the number of agreement attractors.

Probing Dataset We adopt the benchmark by Gulordava et al. (2018), henceforth called G18, which extends the evaluation of Linzen et al. (2016) to more languages and more agreement constructions, automatically harvested from corpora using POS patterns. G18 also introduced two conditions to test whether a model relies on semantic cues or purely grammatical knowledge to predict agreement:

1. Original : Sentences automatically extracted from corpora;
2. Nonce : Nonsensical but grammatical sentences created by randomly replacing all content words in the original sentence with random words with same morphological class.

Thus, this is one of few existing tasks that allow us to measure the transfer of grammatical knowledge in isolation. Using the G18 benchmark, we compare mNNs with monolingually trained models, in order to compare if the addition of a related language improves the long-range agreement accuracy of the monolingual model. We expect this to happen for languages that have the same number agreement patterns, like French and Italian.

Probed Models Similar to G18, we train 2-layer LSTMs with embedding and hidden layer size of 650, for 40 epochs, using a dataset of crawled Wikipedia articles. These language models are trained on next word prediction and do not receive any specific supervision for the syntactic task. $L1$ is our helper language and $L2$ is the target language where we measure agreement accuracy. Fig. 2 shows our different training setups. To simulate a low-resource setup and possibly increase the chances of transfer, we train our bilingual LMs on a shuffled mix of a larger L1 corpus ($L1_{large}$, 80M tokens) and a smaller L2 corpus ($L2_{small}$, 10M tokens). L2 is oversampled to approximately match the amount of L1 sentences. This bilingual model ($LM_{L1+L2_{small}}$) is compared to a baseline monolingual LM trained on a small L2 corpus ($LM_{L2_{small}}$). As upper bound, we also show the results of a model trained on more L2

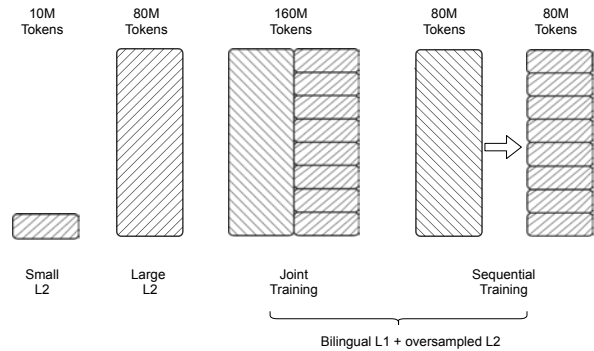


Figure 2: Monolingual and bilingual LM training schemes used in our agreement experiments.

data (80M). This model performs closely to the results reported by G18 with a similar setup.

Most experiments in this paper are performed by *joint training*, i.e. the model is trained on the mixed language data since initialization. However in Sect. 4.2 we also evaluate *pre-training*: i.e. the LM is first trained on L1 data, then after convergence, it continues training on L2 data (see Fig. 2). A language tag is introduced at the beginning of each sentence. The vocabulary for each language consists of the 50k most frequent tokens, with the remaining tokens replaced by the unknown tag. The bilingual vocabulary is the union of the language-specific vocabularies, resulting in a total of 88k words in our main language pair (French-Italian). In Sect. 4.2 we compare this setup (called *natural overlap*) to a *no-overlap* setup where all words are prepended with a language tag, resulting in a bilingual vocabulary of 100k words.

3.2 Cross-lingual Syntactic Category Classification

To verify whether basic syntactic categories are shared among different language representations in mNNs, we inspect the activations of our trained LMs when processing a held-out corpus. Specifically we build linear classifiers to predict either the PoS tag or the Dependency label (type of relation to the head) of a word from its hidden layer representation. This setup is similar to previous work (Blevins et al., 2018; Tenney et al., 2019b), however our diagnostic classifiers are trained on L1 and tested on L2.¹ If syntactic categories are shared, we expect to see minor drops in classification accuracy compared to a classifier trained and

¹Another difference regards the dependency classification: Blevins et al. (2018) uses constituency parsing and Tenney et al. (2019b) predicts dependency arcs given word *pairs*.

tested on $L2$. In other words, we ask whether, e.g., French and Italian adjectives or subjects are recognizable by the same NN activations.

Several studies such as (Bisazza and Tump, 2018; Hewitt and Liang, 2019; Pimentel et al., 2020) have criticised diagnostic classifiers for overestimating the ability of neural networks to capture linguistic information. We partly address these pitfalls by comparing classification accuracy on top of our trained mNNs with that of their corresponding randomly initialized counterparts.

Probing Dataset We probe our models on manually annotated coarse-grained PoS and Dependency labels taken from Universal Dependency Treebanks (Nivre et al., 2019). Specifically, we use French-GSD (389k tokens), Italian-ISDT (278k), Spanish-AnCora (548k), and German-GSD (288k). UD sentences are fed to a trained model’s encoder and the resulting last-layer activations are used to build the probing classifiers.

Probed Models We first apply the PoS and Dependency probing tasks to the Wikipedia-based LMs described in Sect. 3.1. To study the effect of training objective (next word prediction *vs* translation to a third language), in Sect. 5.2 we perform another set of controlled experiments using the Europarl² parallel corpus. Our dataset consists of $L1 \rightarrow$ English parallel sentences, where $L1$ is one of nine languages chosen from three different families: French, Italian, Portuguese, Spanish (Romance); German, Dutch, Swedish and Danish (Germanic) and Finnish (Uralic), with about 45.9M tokens for each language pair. The NMT models implement a standard attentional sequence-to-sequence architecture based on 4-layer bidirectional LSTMs (Bahdanau et al., 2015) with embedding and hidden layer size of 1024. To maximize comparability between translation and language modeling objectives, the LMs in these experiments are also 4-layer bidirectional (BiLMs, à la Peters et al. (2018)) with the same hidden layer size, trained on the source-side portion of our Europarl dataset.

3.3 Word Translation Retrieval

To put syntactic transfer in contrast with other types of transfer effects, we also experiment with word translation retrieval (henceforth abbreviated as WTR). This was used as a probing task for

²<http://www.statmt.org/europarl/v7/>

cross-lingual word embeddings in (Lample et al., 2018; Conneau et al., 2018a) and involves calculating the distance (measured by cosine similarity) between the embedding of a source language word (e.g., *bonjour*) and that of its translation (e.g., *buongiorno*). Since the task is context independent, only the word-type embeddings are probed. We interpret precision in this task as a measure of the alignment of two word embedding spaces, that is *lexical-semantic* transfer.

Lexicon The bilingual lexicon from MUSE (Lample et al., 2018) is used as gold standard for this task. MUSE is available for several language pairs and includes polysemous words (many-to-many pairs). For each language pair, we use 1.5k source and 200k target words.

4 Does Exposure to $L1$ Improve Implicit Syntactic Knowledge on a Related $L2$?

To answer RQ1 we use the number agreement task, which is explained in detail in Sect. 3.1. We choose Italian (IT) and Russian (RU) from the G18 dataset as our target languages $L2$. As helper languages, $L1$, we choose French (FR) and Spanish (ES) for $L2$ IT, and FR and Ukrainian (UK) for $L2$ RU, which allows us to study the impact of language relatedness. Accuracy is calculated as follows: for each sentence in the $L2$ benchmark, if the probability of the correct verb form is higher than the incorrect form, the agreement is said to be correct, and incorrect otherwise.

4.1 Main Results

Figure 3 shows the results. In this set of experiments, the bilingual models are trained by *joint training* using the union of the vocabularies in the two languages (*natural overlap*). See also Sect. 3.1. As in (Gulordava et al., 2018), the frequency baseline selects the most frequent word form (singular or plural) for each sentence.

Looking at the Original sentences, we see that the bilingual models outperform the respective small monolingual models in the closely related pairs $ES \rightarrow IT$ (86.8 *vs* 79.8) and $UK \rightarrow RU$ (90.4 *vs* 88.2). However the addition of FR data results in lower accuracies on both $L2$ s. While this was expected in the unrelated pair $FR \rightarrow RU$, the large drop in $FR \rightarrow IT$ is harder to explain.

When semantic cues are removed (Nonce sentences), $ES \rightarrow IT$ is the only bilingual model to outperform its monolingual counterpart (80.7 *vs*

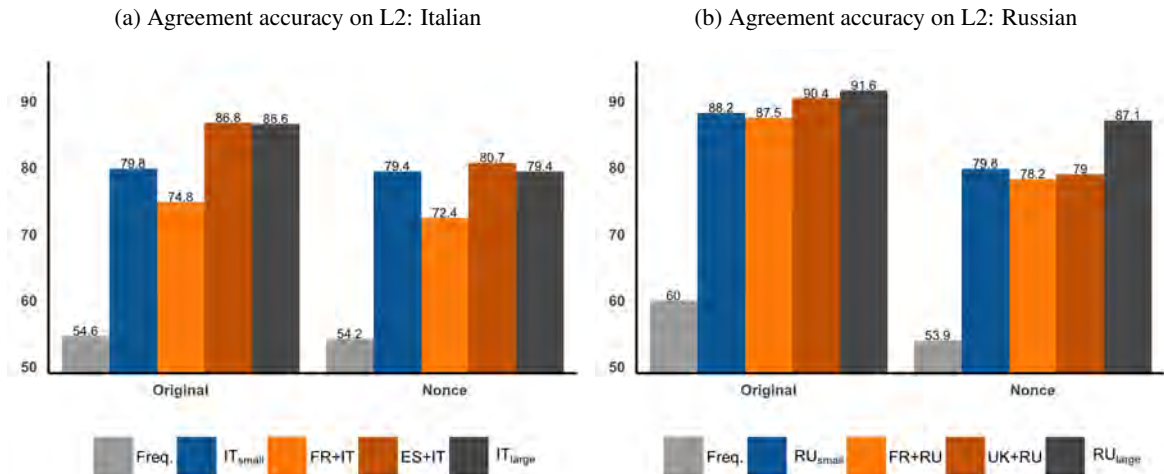


Figure 3: Probing Wikipedia-based monolingual and bilingual LMs on the agreement benchmark of Gulordava et al. (2018). Freq. is the Frequency baseline. Blue and black bars represent small and large L2 models, respectively. Orange bars represent bilingual models.

79.4), while the accuracy drop in FR→IT gets even larger (72.4 vs 79.4). This shows that exposing the model to a related language L1 is not guaranteed to improve implicit syntactic knowledge of L2, even when the rules of number agreement are largely shared between L1 and L2. On the contrary, our experiments suggest that in some cases L1 negatively interferes with the task in L2.

4.2 Effect of Training Regime and Vocabulary Overlap on Agreement

Could transfer in FR→IT be hampered by some of our experimental choices? To consolidate our findings, we experiment with a different training regime (*pre-training*) and a different vocabulary construction method (*no-overlap*). As shown in Table 1, both training regime and vocabulary overlap have a visible effect on the transfer of syntactic knowledge between FR and IT. Pre-training considerably reduces the negative interference effect observed in joint training, and even leads to a higher accuracy on Original sentences in the no-overlap setup (83.2 vs 79.8). Eliminating vocabulary overlap (None) also leads to better agreement scores in most cases. The best gain overall is obtained by the jointly trained model with no overlap (85.7 vs 79.8) in the Original sentences, whereas no gain is observed in the Nonce sentences.

In summary, we find limited and inconsistent evidence of transfer of purely grammatical knowledge in our bilingual models. Also contrary to our expectations, sharing more parameters (natu-

ral overlap) and mixing languages since the beginning of training leads to more negative interference than positive transfer in the FR-IT pair.

IT _{small}	Bilingual (FR+IT _{small})				IT _{large}	
	Joint Training		Pre-Training			
	Natural	None	Natural	None		
Original	79.8	74.8	85.7	79.8	83.2	86.6
Nonce	79.4	72.4	77.6	77.7	76.8	79.4

Table 1: Impact of training regime and vocabulary overlap on agreement accuracy (FR→IT).

5 Do mNNs Induce Shared Syntactic Categories?

Predicting long-range agreement is a rather complex task: in principle, besides learning agreement rules, the model has to discern several syntactic categories such as number, PoS and dependencies (e.g. distinguishing subject from other noun phrases). In practice, previous work (Ravfogel et al., 2018) showed that LSTMs sometimes resort to shallow heuristics when predicting agreement.

In this section we therefore investigate whether our mNNs induce at least basic syntactic categories that are shared across languages (RQ2). We assume this is a necessary condition to enable transfer of purely grammatical knowledge, like agreement in nonce sentences, and beyond.

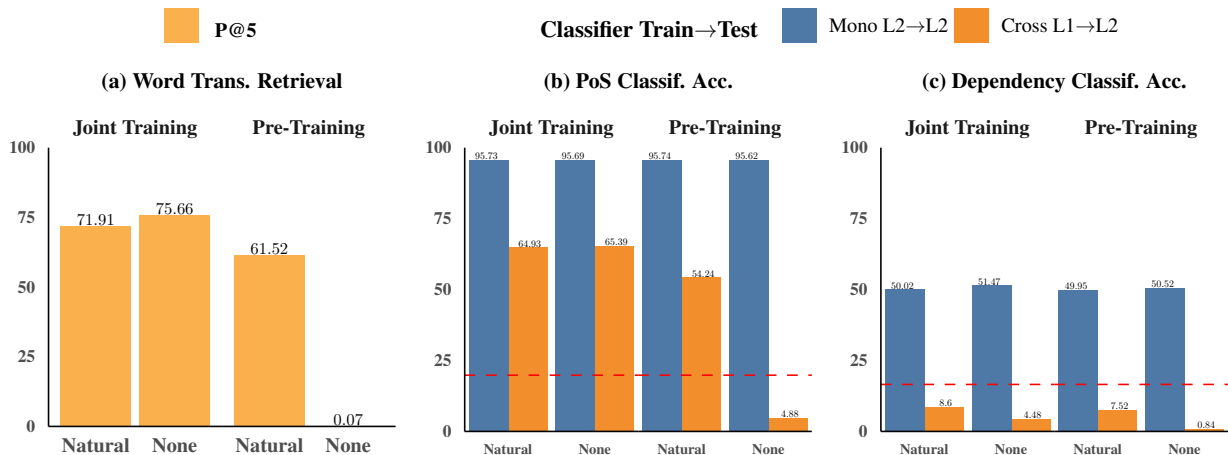


Figure 4: Semantic *vs* syntactic transfer in Wikipedia-based FR-IT bilingual LMs: (a) Word translation retrieval precision (P@5) measures lexical-semantic transfer; (b) PoS accuracy and (c) Dependency accuracy measure syntactic transfer. The classifiers are always tested on L2 (IT), and trained on either L2 or L1 (FR). If syntactic categories were perfectly shared across languages, we should observe no drop between the blue and orange bars. Dashed red lines show majority baselines for both (b) and (c).

5.1 Effect of Training Regime and Vocabulary Overlap on Syntactic Category Transfer

In this section we examine the same FR-IT Wikipedia-based LMs described in section 4.2. Figure 4(a) shows that joint training yields better alignment of the word embedding spaces compared to the pre-training setup, which confirms the findings by Ormazabal et al. (2019). Secondly, eliminating vocabulary overlap does not necessarily imply less alignment. Interestingly, work on m-BERT/XLM models has also shown that vocabulary overlap has a much smaller effect on transfer than previously believed (Wu et al., 2019). An exception to this is the combination of pre-training and disjoint vocabulary (dubbed P/D), which gives near zero alignment of both lexical and syntactic spaces. This suggests that sharing hidden layers is not a sufficient ingredient to adapt a pre-trained model on a new (even if related) language, and that specific techniques should be used when joint training is not a viable option (Wang et al., 2019; Artetxe et al., 2019).

Moving to the transfer of syntactic categories (Fig. 4(b)) we find that all cross-lingually trained PoS classifiers (except P/D) perform much better than the majority baseline but notably worse than the corresponding monolingually trained classifiers. As for dependency classification (Fig. 4c), accuracies are low overall and no cross-lingual

classifier outperforms the majority baseline. In summary, some form of syntactic transfer indeed occurs, but truly language-agnostic syntactic categories (such as *noun* or *subject*) have not emerged in our mNN representations.

5.2 Training Objective, Number of Input Languages, and Language Relatedness

We now study whether a different training objective, namely translation to a third language (English), leads to more syntactic transfer among input languages. We also check whether number of input languages and language relatedness play a significant role in the sharing of syntactic categories. All models in this section are jointly trained with natural vocabulary overlap on Europarl, and compared to their randomly initialized equivalents following Zhang and Bowman (2018). Dependency classification results are omitted as they were always below the majority baseline.

Learning Objective As shown in Fig. 5(a,b), the translation objective has a slightly negative impact on the alignment of word embedding spaces when all other factors are fixed. The translation objective also leads to lower PoS accuracy (monolingually probed), confirming previous results by Zhang and Bowman (2018). However, translating to English does result in visibly better cross-lingual transfer of PoS categories (mono/cross-lingual drop of -27.7 for translation *vs* -37.2 for

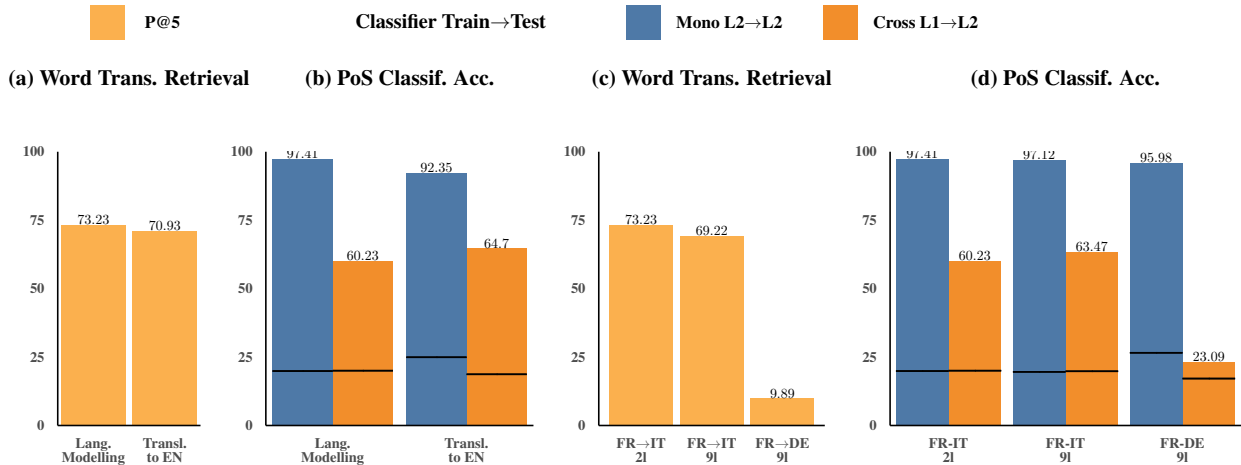


Figure 5: Semantic (word translation retrieval) vs syntactic (PoS classif.) transfer in Europarl-based bidirect. mNNs. (a,b) Effect of training objective: next word prediction vs translation to English. (c,d) Effect of number of input languages (2 vs 9) and language relatedness (FR-IT vs FR-DE) for the bidi-LM objective. Horizontal lines (b,d) refer to the corresponding randomly initialized mNNs.

language modelling), showing that what are optimal conditions for lexical-semantic may no be optimal for syntactic transfer.

Number of Source-side Languages For the remaining experiments we look at the (bidirectional) LM objective. As shown in Fig. 5(c,d), moving from 2 input languages to 9 results in lower WTR precision but higher cross-lingual PoS accuracy. This suggests that adding more languages does not cause mNN representations to lose syntactic information and actually leads to more sharing of syntactic categories across languages. The generality of this remark is however restrained by our findings on language relatedness.

Language Relatedness Fig. 5(c,d) also shows that moving from a very related pair of input languages (FR-IT) to a less related one (FR-DE) results in dramatically lower transfer of both lexical-semantics and syntactic categories. To substantiate this finding, we extend the analysis of our 9-language LM to more training-test pairs (we select a subset of languages for which a sizeable UD treebank exists). The results in Fig. 6 confirm that, for both lexical-semantics and syntax, the related languages FR, IT and ES report considerably higher values than those involving DE, while the smallest drop (-6.45) is seen between FR→FR and FR→IT. While we expected transfer to depend on relatedness, we did not expect the effect to be so large given that DE is not completely

unrelated from the Romance languages.

6 Conclusions

We have presented an in-depth analysis of various factors affecting cross-lingual syntactic transfer within multilingually trained LSTM-based language (and translation) models. Our main result is a negative one: Transfer of purely grammatical knowledge (specifically long-range agreement in nonce sentences) is very limited in general – confirming recent findings by Mueller et al. (2020) – and strongly dependent on the specific choice of source-target languages. Namely, small gains were only reported on ES→IT, while a considerable drop was reported on FR→IT and almost no change was reported on UK→RU. When semantic cues were not removed (original sentences), transfer levels were overall higher with a peak of +7% absolute in ES→IT, but FR→IT still suffered a considerable loss (-5%). While ES is arguably closer to IT than FR, we cannot yet find a convincing linguistic explanation for the large differences observed. Our second set of experiments shows that POS categories are shared to a moderate extent, but dependency categories are not shared at all in our models. This suggests that syntactic knowledge transfer within our multilingual models is rather shallow, and may explain the lack of agreement transfer.

Our experiments with different training objectives and number of input languages show that

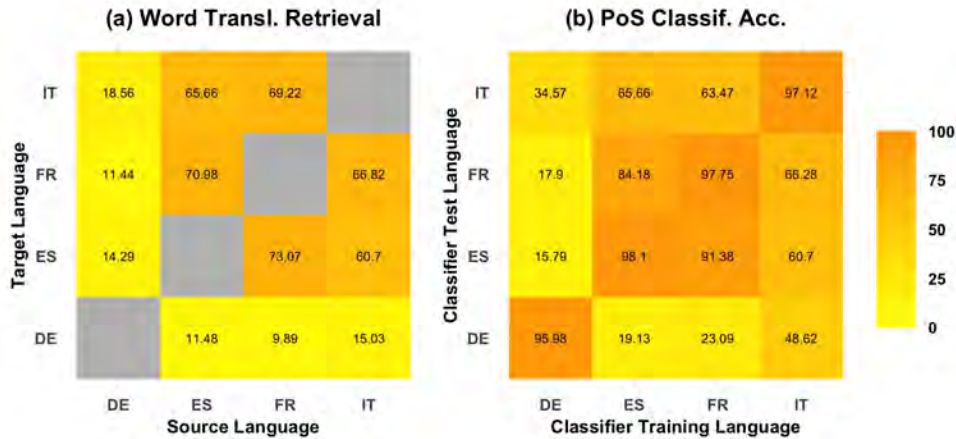


Figure 6: Pairwise semantic and syntactic transfer in the 9-language bidi-LM (a subset of languages is shown). Non-applicable (monolingual) settings in (a) are greyed out. Diagonal values in (b) are scores of monoling. L2→L2 classifiers, while remaining values are for cross-ling. L1→L2 ones.

what are optimal conditions for the alignment of word embedding spaces (lexical-semantic transfer) may not be optimal for syntactic transfer, and vice versa. Language relatedness is by far the most determining factor for both word embedding alignment and POS transfer. And finally, scaling from two languages to a mix of nine languages from three different families results in better POS transfer between related languages but considerably worse between unrelated ones. Together with the findings by Wu et al. (2019), our results suggest that scaling to highly multilingual models may improve syntactic transfer among the most related languages by decreasing the per-language capacity, but may also exacerbate the divergence among less related ones. Thus modern multilingual NNs appear still far from acquiring a true interlingua.

Acknowledgements

Arianna Bisazza was partly funded by the Netherlands Organization for Scientific Research (NWO) under project number 639.021.646.

References

Roei Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Wasi Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard

Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2440–2452, Minneapolis, Minnesota. Association for Computational Linguistics.

Joakim Nivre et al. 2019. Universal dependencies 2.4. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*,

- ICLR 2015, San Diego, CA, USA, May 7-9, 2015, *Conference Track Proceedings*.
- Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.
- Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.
- Arianna Bisazza and Clara Tump. 2018. The lazy encoder: A fine-grained analysis of the role of morphology in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2871–2876. Association for Computational Linguistics.
- Johannes Bjerva and Isabelle Augenstein. 2018. From phonology to syntax: Unsupervised linguistic typology at different levels with language embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 907–916, New Orleans, Louisiana. Association for Computational Linguistics.
- Terra Blevins, Omer Levy, and Luke Zettlemoyer. 2018. Deep RNNs encode soft hierarchical syntax. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19, Melbourne, Australia. Association for Computational Linguistics.
- Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual bert.
- Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7057–7067. Curran Associates, Inc.
- Alexis Conneau, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018a. Word translation without parallel data. In *International Conference on Learning Representations (ICLR)*.
- Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018b. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Jacob Devlin. 2018. Multilingual BERT Readme Document. <https://github.com/google-research/bert/blob/a9ba4b8d7704c1ae18d1b28c56c0430d41407eb1/multilingual.md>.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.
- Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.
- Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.
- Robert J. Hartsuiker, Martin J. Pickering, and Eline Veltkamp. 2004a. Is syntax separate or shared between languages?: Cross-linguistic syntactic priming in spanish-english bilinguals. *Psychological Science*, 15(6):409–414. PMID: 15147495.
- Robert J. Hartsuiker, Martin J. Pickering, and Eline Veltkamp. 2004b. Is syntax separate or shared between languages?: Cross-linguistic syntactic priming in spanish-english bilinguals. *Psychological Science*, 15(6):409–414.
- John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

- Scott Jarvis and Anna Pavlenko. 2008. *Crosslinguistic influence in language and cognition*. Routledge.
- Ganesh Jawahar, Benoît Sagot, and Djamel Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Eric Kellerman and ed. Sharwood Smith, Michael. 1986. *Crosslinguistic influence in second language acquisition*. Pergamon.
- Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.
- Gerrit Jan Kootstra, Janet G. Van Hell, and Ton Dijkstra. 2012. Priming of code-switches in sentences: The role of lexical repetition, cognates, and language proficiency. *Bilingualism: Language and Cognition*, 15(4):797–819.
- Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.
- Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.
- Jindrich Libovický, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *CoRR*, abs/1911.03310.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.
- Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.
- Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.
- Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 629–637.
- Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.
- Terence Odlin. 1989. *Language Transfer: Cross-Linguistic Influence in Language Learning*. Cambridge Applied Linguistics. Cambridge University Press.
- Aitor Ormazabal, Mikel Artetxe, Gorka Labaka, Aitor Soroa, and Eneko Agirre. 2019. Analyzing the limitations of cross-lingual word embedding mappings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4990–4995, Florence, Italy. Association for Computational Linguistics.
- Robert Östling and Jörg Tiedemann. 2017. Continuous multilinguality with language vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 644–649, Valencia, Spain. Association for Computational Linguistics.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.
- Shauli Ravfogel, Yoav Goldberg, and Francis Tyers. 2018. Can LSTM learn to capture agreement? the case of Basque. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 98–107, Brussels, Belgium. Association for Computational Linguistics.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.
- Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 477–487.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4593–4601.
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.
- Kristen M. Tooley and Matthew J. Traxler. 2010. Syntactic priming effects in comprehension: A critical review. *Language and Linguistics Compass*, 4(10):925–937.
- Ke Tran and Arianna Bisazza. 2019. Zero-shot dependency parsing with pre-trained multilingual sentence representations. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 281–288, Hong Kong, China. Association for Computational Linguistics.
- Marina Vasilyeva, Heidi Waterfall, Perla B. Gámez, Ligia E. Gómez, Edmond Bowers, and Priya Shimpi. 2010. Cross-linguistic syntactic priming in bilingual children. *Journal of Child Language*, 37(5):1047–1064.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.
- Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. 2019. Cross-lingual BERT transformation for zero-shot dependency parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5720–5726, Hong Kong, China. Association for Computational Linguistics.
- Shijie Wu, Alexis Conneau, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Emerging cross-lingual structure in pretrained language models. *CoRR*, abs/1911.01464.
- Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.
- Kelly Zhang and Samuel Bowman. 2018. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361, Brussels, Belgium. Association for Computational Linguistics.
- Yuan Zhang and Regina Barzilay. 2015. Hierarchical low-rank tensors for multilingual transfer parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1857–1867.

Speaker Verification Experiments for Adults and Children using a shared embedding spaces

Tuomas Kaseva, Hemant Kathania, Aku Rouhe, Mikko Kurimo

Department of Signal Processing and Acoustics, Aalto University, Finland

(firstname.lastname)@aalto.fi

Abstract

In this work, we present our efforts towards developing a robust speaker verification system for children when the data is limited. We propose a novel deep learning -based speaker verification system that combines long-short term memory cells with NetVLAD and additive margin softmax loss. First we investigated these methods on a large corpus of adult data and then applied the best configuration for child speaker verification. For children, the system trained on a large corpus of adult speakers performed worse than a system trained on a much smaller corpus of children’s speech. This is due to the acoustic mismatch between training and testing data. To capture more acoustic variability we trained a shared system with mixed data from adults and children. The shared system yields the best EER for children with no degradation for adults. Thus, the single system trained with mixed data is applicable for speaker verification for both adults and children.

Index Terms: additive margin softmax loss, NetVLAD aggregation, recurrent neural network, speaker verification for children.

1 Introduction

The use of speaker verification (SV) technology for children has many beneficial application areas, such as child security and protection, entertainment, games and education. For example, in an interactive class the teacher could identify each child, by continuing a previous lecture and adapt its content with the child’s speech, and log the child’s responses without a conventional login process (Safavi et al., 2018, 2012).

The acoustic and linguistic characteristic of children’s speech differ from adults’ speech (Lee et al., 1999). The main differences are in pitch, speaking rate and formant frequencies (Kumar Kathania et al., 2020; Shahnawazuddin et al., 2019). These acoustic differences together with the lack of training data make SV more challenging. Little work has been reported in this area. In (Shahnawazuddin et al., 2020) in-Domain and out-of-Domain data augmentation are used to improve a child SV system in a limited data scenario. In (Safavi et al., 2012) vocal tract information is used for children’s SV. Explanation for degraded recognizer scores through acoustic changes resulting from voice disguise is presented in (González Hautamäki et al., 2019).

In this work, we explore how recent advances in (adult) SV could aid in child SV, as well. In particular, we combine adult and child SV into a single task, by using a shared embedding space for adult and child speakers. This allows us to leverage the large resources available for adult speakers for the low-resource child speaker verification task. In applications where both adult and child speakers can be expected, it is also natural to use a single shared system for both groups; we find that a shared system can be used which benefits child SV without degrading adult SV performance.

Contributions. We construct a neural SV system, which leverages recent advancements in the field. In particular, we find improvements from using the additive-margin softmax loss and the NetVLAD time aggregation methods. In contrast to most recent literature, which uses convolutional neural layers, we apply recurrent layers, motivated by success in speaker diarization (Kaseva et al., 2019). We compare our results to recent high-performing systems of similar complexity. Though we do not outperform the top results, the comparison validates our approach. We then apply the proposed SV system to adult and child SV

and find that using a shared embedding for both adult and children improves child SV drastically without affecting adult SV performance.

2 Related speaker verification work

In recent years, deep learning motivated approaches have shown significant progress in SV. We consider three main reasons for their success. Firstly, larger and more realistic speakers-in-the-wild speaker recognition datasets have become available to the public (Nagrani et al., 2017; Chung et al., 2018; McLaren et al., 2016). Secondly, the loss functions used in the training of neural networks have advanced. In general, the main objective of the neural networks designed for SV is to transform a given recording into a speaker embedding which embodies the speaker characteristics of the recording (Snyder et al., 2018, 2019; Bredin, 2017; Li et al., 2017). In the most current methods, the embeddings are learned in a speaker identification process, where original softmax loss is modified by adding a margin to the class decision boundaries (Liu et al., 2019; Xie et al., 2019; Xiang et al., 2019). This allows efficient training and reduces the intra-class variance of the created embeddings (Wang et al., 2018a; Deng et al., 2019; Liu et al., 2017). Finally, the neural network architectures have developed. One of the most prominent discoveries has been x-vectors, speaker embeddings which are extracted from an architecture based on time-delay neural networks (TDNNs) (Snyder et al., 2018; Liu et al., 2019; Xiang et al., 2019). X-vectors have been shown to outperform i-vectors, which have enjoyed a state-of-the-art status in SV for a long time (Dehak et al., 2010). In some cases, i-vectors have also been inferior to the SV systems which utilize convolutional neural networks (CNNs) (Chung et al., 2018; Ravanelli and Bengio, 2018). Furthermore, novel aggregation methods for neural networks have been proposed. Whereas average pooling has been used extensively before, the most recent approaches include statistics pooling, attentive statistics pooling and NetVLAD (vector of locally aggregated descriptors) (Okabe et al., 2018; Arandjelovic et al., 2016; Xie et al., 2019).

In addition, recurrent neural networks (RNNs) with long-short term memory (LSTM) cells (Hochreiter and Schmidhuber, 1997) have been experimented with (Wan et al., 2018; Bredin, 2017; Heigold et al., 2016). Most importantly,

they have shown success in a related task, online speaker diarization (Wang et al., 2018c; Zhang et al., 2019; Wisniewski et al., 2017). In this task, LSTMs have been able to create compact speaker embeddings from very short segments.

Our approach has some similarities with Wan et al. (Wan et al., 2018). As in their work, we use LSTMs in sliding windows. However, unlike them, we do not apply the generalized end-to-end loss for neural network training. Instead, we use the AM-softmax loss (Wang et al., 2018a). Furthermore, unlike them, we combine LSTMs with NetVLAD. Although NetVLAD layer has been previously used for SV (Xie et al., 2019), in that study, the layer was connected to a CNN. NetVLAD has been originally designed for aggregation of CNNs (Arandjelovic et al., 2016) and to the best of our knowledge, we are the first to use it with LSTMs in any application.

3 Proposed methods

In this section, we detail our SV system which consists of three stages: splitting, embedding and averaging, as illustrated in Fig. 1.

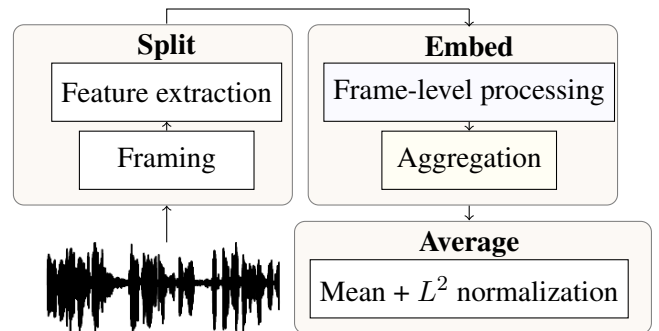


Figure 1: Schematic of our speaker embedding extraction approach.

Split. First, the audio input is split into overlapping windows with short, roughly 2 seconds or less, duration. Time-varying features are then extracted from each frame, resulting in a set of feature sequences \mathbf{x} . The sequences consist of 30 Mel-Frequency Cepstral Coefficients (MFCC) which are extracted every 10ms with 25ms frame length. Every \mathbf{x} is normalized with zero mean and unit variance.

Embed. In the next step, each \mathbf{x} is transformed into a speaker embedding. This can be further divided into two distinct steps: frame-level processing and aggregation.

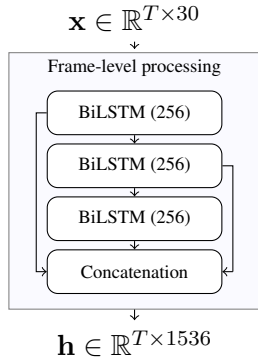


Figure 2: Frame-level processing. The numbers refer to the number of hidden units in each layer.

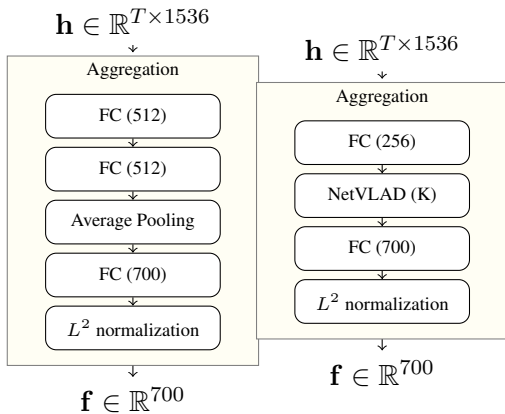


Figure 3: Two different aggregation approaches: average pooling on the left and NetVLAD on the right. FC refers to a fully connected layer and the numbers to the output dimensionality.

In frame-level processing, each \mathbf{x} is transformed into higher level frame-features \mathbf{h} . In our approach, \mathbf{x} is fed to a cascade of three bidirectional LSTM layers with skip connections. Each layer outputs the hidden states of both the forward and backward LSTMs. These outputs are concatenated resulting in \mathbf{h} as illustrated in Figure 2. The structure of the cascade adheres to (Wisniewski et al., 2017). A more common choice for frame-level processing blocks is to use convolutional layers.

In aggregation, the higher level features \mathbf{h} are compressed into a speaker embedding \mathbf{f} . We compare two aggregation approaches: average pooling and NetVLAD. The aggregation components are illustrated in Fig. 3. Note that the aggregation component with average pooling has a slightly different configuration than its NetVLAD motivated counterpart. This choice was based on balancing the number of parameters in both neural networks.

We force the embeddings to be L^2 normalized in both components. As a result, cosine distance is the most natural distance metric between different embeddings. A rectified linear unit activation is used in all of the fully connected (FC) layers. We also apply batch normalization (Ioffe and Szegedy, 2015) after each layer except L^2 normalization layers. This means that the last two layers of both components perform normalization. Although this might seem strange, we discovered it to be beneficial in the preliminary experiments.

The operation of the NetVLAD layer can be summarized as follows. Let us denote the output of the preceding FC layer as $\mathbf{v} \in \mathbb{R}^{T \times 256}$. First, \mathbf{v} is transformed into $\mathbf{V} \in \mathbb{R}^{K \times 256}$ according to a formula (Arandjelovic et al., 2016)

$$\mathbf{V}(k, d) = \sum_{t=1}^T \frac{e^{\mathbf{w}_k^T \mathbf{v}_t + b_k}}{\sum_{k'=1}^K e^{\mathbf{w}_{k'}^T \mathbf{v}_t + b_{k'}}} (\mathbf{v}_{td} - \mathbf{c}_{kd}), \quad (1)$$

where $\mathbf{c} \in \mathbb{R}^{K \times 256}$, $\mathbf{w} \in \mathbb{R}^{K \times 256}$ and $\mathbf{b} \in \mathbb{R}^K$ are learnable parameters. In this formulation, \mathbf{c} can be interpreted as a set of K cluster centers which characterize the distribution of \mathbf{v} (Xie et al., 2019). More specifically, \mathbf{V} consists of first order statistics of residuals $\mathbf{v}_d - \mathbf{c}_k$ in which each element is weighted based on \mathbf{v} and the cluster index k . The number of clusters K is given as an input to the layer. After calculation of the residuals, each row of \mathbf{V} is first L^2 normalized and then concatenated resulting in $\mathbf{V}_f \in \mathbb{R}^{256 * K}$. In the literature, additional L^2 normalization operation has been applied after flattening (Xie et al., 2019; Arandjelovic et al., 2016). However, we use batch normalization instead. We found this normalization to perform generally better in the preliminary experiments. The use of NetVLAD in this study is motivated by its recent success in SV when combined with CNNs (Xie et al., 2019). Here, we show that NetVLAD is beneficial also with LSTMs.

Average. In the final stage, we compute a single embedding $\mathbf{f}_c \in \mathbb{R}^{700}$ for the recording by averaging the created speaker embeddings and L^2 normalizing the average. When considering \mathbf{f}_{c1} and \mathbf{f}_{c2} extracted from two different recordings, our system performs SV by computing cosine distance between the embeddings and by thresholding the obtained value. Another popular method for comparing the embeddings is Probabilistic Discriminant Analysis (PLDA) (Ioffe, 2006; Snyder et al.,

2018). PLDA could result in performance improvements (Liu et al., 2019), but also increase the complexity of our system, and we do not apply it in this work.

4 Experiments

Data. We use two training sets for adult speakers which are both generated from Voxceleb2 (Chung et al., 2018). In the first, abbreviated as $VC2$, all recordings in Voxceleb2 are windowed into 2 second samples with 1 second overlap. The reason for this choice is the training objective of our neural networks that is to identify a speaker from a given training set based on a 2 second segment of speech. The duration was not selected arbitrarily: we experimented also with setting it to 1 and 2.5 seconds. The former was too short for neural networks to learn speaker characteristics properly and the latter did not generally improve the performance of the networks. $VC2$ consists of roughly 6.83 million training samples from 5994 speakers.

The second set, $VC2_C$, is otherwise the same as $VC2$ but excludes a portion of the samples based on a heuristic cleaning algorithm. The motivation for this algorithm came from our listening tests which confirmed that Voxceleb2 included wrongly labeled speaker identities in some cases. The exclusions removed approximately 46k samples from $VC2$ but retained the number of speakers, 5994. Given samples S_i belonging to i -th speaker in $VC2$, the cleaning algorithm operates in four steps:

1. Create a speaker embedding \mathbf{f} for each sample in S_i .
2. Cluster the embeddings with spherical K-means setting $K = 2$ into groups G_1 and G_2 .
3. Calculate the average of silhouette coefficients ϕ of the clustering result. Further details of these coefficients are given in (Rousseeuw, 1987).
4. If $|G_1| > 0.6|G_1 \cup G_2|$ and $\phi > 0.3$, exclude all samples belonging to G_2 from the training set. Here, $|G_i|$ refers to a number of elements in group G_i .

In summary, the algorithm investigates whether the recordings initially assigned to a single speaker might contain also another speaker. The algorithm

removes samples from S_i only if the speech material portions of the clusters are not balanced and if the clustering result has a high reliability. This reliability is measured using silhouette coefficients. Speaker embedding extraction was performed using an initial neural network which has the same average pooling based architecture as described in the previous section, but was trained only with 4000 speakers from $VC2$.

We evaluate our models also using the cleaned versions of Voxceleb1 verification test sets, Voxceleb1-test (VC_t), Voxceleb1-H (VC_H) and Voxceleb1-E (VC_E) (Chung et al., 2018). The recordings in these sets are framed to 2 second duration segments with 1.5 seconds overlap. The overlap duration was determined in the preliminary experiments.

We construct also our own verification set from the development set of Voxceleb1. This set is used for model evaluation during training. The set consists of speech segments with a fixed 2 seconds duration, and which each are extracted from a unique session and speaker. The number of extracted segments is close to 20k and they belong to 1211 speakers. We form close to 150k segment pairs where half of the pairs correspond to the same speaker and the other half to different speakers. We name this verification set as VC_{2sec} . The set is disjoint in speakers with VC_t but not with VC_H and VC_E . However, we consider this evaluation set to be valid since the pair compositions and segment durations of VC_{2sec} differ significantly from VC_H and VC_E .

For child speech experiments we used CSLU kids (Khaldoun Shobaki, 2007) database for training. It has 1110 speakers of English language with age range from 6 to 16 years and sampling rate 16 kHz. For testing the system we used PF-STAR (Batliner et al., 2005) and the Finnish SpeechDat (Rosti et al., 1998) datasets. PFSTAR has 134 speakers of English with age range from 4 to 14 years, originally sampled at 22,050 Hz. The down-sampling at 16 kHz was performed for consistency with the model. SpeechDat has 354 speakers of Finnish with age range from 6 to 14 years, originally data sampled at 8 kHz. The up-sampling at 16 kHz was performed for consistency. For children’s experiments we used the same speaker embedding method as adults.

Training. In training, the output of the aggregation component is connected to a fully connected

layer which is used for a speaker identification task. Training has two stages: warm-up with the softmax loss and fine-tuning with the AM-softmax loss (Wang et al., 2018a). In the warm-up, the neural network is trained for 5 epochs, using Adam optimizer with 0.01 learning rate. Batch size is chosen as 512. We generally observed that the performance of the neural networks on the VC_{2sec} would not improve after the fifth epoch when using the softmax loss.

In the fine-tuning, the softmax loss for i -th training sample is reformulated as

$$L_i = \log \frac{e^{s(\mathbf{W}_{y_i}^T \mathbf{f} - m)}}{e^{s(\mathbf{W}_{y_i}^T \mathbf{f} - m)} + \sum_{j=1, j \neq y_i}^{5994} e^{s\mathbf{W}_j^T \mathbf{f}}}, \quad (2)$$

where y_i is the label of i -th training sample, $\mathbf{W} \in \mathbb{R}^{700 \times 5994}$ a learnable weight matrix with all rows L^2 normalized and s and m a given scale and margin. Equation 2 is known as the AM-softmax loss (Wang et al., 2018a). We set $m = 0.15$ and $s = 0.25$ based on our preliminary experiments. \mathbf{W} is initialized with the weights of the best neural network configuration found in the warm-up.

The main point of using the AM-softmax loss is to decrease intra-class variance, which is generally difficult with the softmax loss (Wang et al., 2018a; Deng et al., 2019; Liu et al., 2017). In other words, the higher the margin m is set, the more closer, in terms of cosine distance, the speaker embeddings belonging to the same class are forced. The cosine distance metric arises from the L^2 normalizations of both \mathbf{f} and the rows of \mathbf{W} . The scale of s is generally set to a some high value to ensure convergence (Wang et al., 2018b). In recent years, the AM-softmax loss and other similar methods (Liu et al., 2017; Deng et al., 2019) have emerged as state-of-the-art approaches in speaker verification (Xie et al., 2019; Liu et al., 2019; Xiang et al., 2019; Li et al., 2018).

The fine tuning is continued for 10 epochs with otherwise the same setting as in warm-up. We monitor the progress of the training by first computing cosine distances between the embeddings of each pair in VC_{2sec} and then calculating equal error rate (EER) on these distances after each epoch. EER is a standard error metric in speaker verification (Snyder et al., 2018; Chung et al., 2018; Xie et al., 2019). Although the VC_{2sec} contains over 150k pairs, the evaluation on this set is efficient during the training since it consists of short, equal length segments which can be embed-

ded rapidly. We save the weights of the neural network after each epoch, and choose the configuration with the best EER value as our final model.

5 Results

First in section 5.1 we validate our SV approach on adult speech. Then in section 5.2 we apply the system with children.

5.1 Validation experiments with adults

In this section, we first investigate the effect of the cleaning algorithm, aggregation and the AM-softmax loss. Finally, we present a results comparison. We use EER as an evaluation metric in all experiments.

Table 1: Effect of training set cleaning (EER %). $K = 30$.

Aggregation	Training set	VC_t	VC_E	VC_H	VC_{2sec}
NetVLAD	VC_2	2.49	2.47	4.53	6.65
NetVLAD	VC_{2C}	2.18	2.45	4.45	6.66

Effect of dataset cleaning. In Table 1, we show that small improvements can be achieved by removing some training data with the cleaning algorithm. This proves that the algorithm is reasonable and also encourages discussion whether some cleaning operation is needed for Voxceleb2. However, the improvements in VC_E and VC_H are minor and with VC_{2sec} , the cleaning has not been beneficial.

Table 2: Effect of K and aggregation (EER %). The training set is VC_{2C} .

Aggregation	K	VC_t	VC_E	VC_H	VC_{2sec}
Average pooling	-	2.46	2.45	4.42	7.05
NetVLAD	8	2.41	2.40	4.35	6.92
NetVLAD	14	2.32	2.37	4.36	6.68
NetVLAD	30	2.18	2.45	4.45	6.66

Effect of aggregation approach. Table 2 investigates the performance of the two aggregation approaches and the choice of K . The results show that NetVLAD is the better approach. This is particularly clear with VC_{2sec} . However, the best scores with different test sets are all obtained with different K values. This result highlights the importance of using multiple different test sets for model evaluation. Nevertheless, we can decide on the best model based on the average over all EER scores. In this case, the NetVLAD-based aggregation with $K = 14$ has the best performance.

Table 3: Effect of loss function (EER %). $K = 30$ and the training set is $VC2_C$.

Aggregation	Loss	VC_t	VC_E	VC_H	VC_{2sec}
NetVLAD	Softmax	3.25	3.30	5.90	8.40
NetVLAD	AM-softmax	2.18	2.45	4.45	6.66

Effect of loss function. Table 3 illustrates that the AM-softmax loss brings significant improvements over the softmax loss. Similar results were obtained with the average pooling aggregation. However, we want to emphasize the results with the NetVLAD aggregation since in (Xie et al., 2019), the use of NetVLAD with the AM-softmax loss has not resulted in notable performance improvements. Here, we demonstrate that the two can be combined successfully. The results with different K values were essentially the same.

Table 4: Results comparison (EER %).

System	Scoring	VC_t	VC_E	VC_H
Xie <i>et al.</i> (Xie et al., 2019)	Cosine	3.22	3.13	5.06
Xiang <i>et al.</i> (Xiang et al., 2019)	PLDA	2.69	2.76	4.73
Ours	Cosine	2.32	2.37	4.36
Zhou <i>et al.</i> (Zhou et al., 2019)	<i>Unknown</i>	2.23	2.18	3.61
Zeinali <i>et al.</i> (Zeinali et al., 2019)	Cosine	1.42	1.35	2.48

Results comparison. In Table 4, we compare our system to other high-performing speaker verification systems. The comparison of our system with the first, x-vector based (Xiang et al., 2019) system and the second, CNN-based (Xie et al., 2019) system is straight-forward since all the systems are trained with the same dataset, Voxceleb2, and because the number of parameters are close to each other: 4.2 million in (Xiang et al., 2019), 7.7 million in (Xie et al., 2019) and 6.7 million in our system. Zhou *et al.* (Zhou et al., 2019) report better results than ours, but they use data augmentation, and do not report the number of parameters used. The current state-of-the-art (single-system) results by Zeinali *et al.* (Zeinali et al., 2019) in the VoxCeleb Speaker Recognition Challenge 2019 leverage data augmentation and more parameters. Our results do not outperform the best published results, but the results still validate our approach, as our results outperform the strong results from (Xie et al., 2019) and (Xiang et al., 2019), which use similar parameter and data constraints.

5.2 Evaluation experiments with children

In the previous section, we presented the effect of dataset cleaning, aggregation approach, and loss function on adult speakers. In this section, we took the best combination of all these for child speech experiments. Details of databases used for the experiments with children is given in section 4.

Table 5: Results on child speakers (EER %).

Training data	PF-STAR	Speechdat	VC_E	VC_H
Adults' $VC2_C$	2.58	10.68	2.37	4.36
Children's CSLU	2.05	10.08	–	
Adults' + Children's	1.12	8.82	2.34	4.39

Table 5 illustrates the performance on child speakers in English and Finnish languages when directly using the adults' model. We also trained a similar model on child speech and report the results in the same table. Based on these results it can be noted that when the system is only trained with adults the performance is lower compared to the system trained only with children even though the children have less training data. To capture more acoustic variability of speakers we trained a shared system with mixed data of both adults and children and tested it with English and Finnish children. The last result of table 5 illustrates that the shared model outperforms both the adult-only and the child-only models for both English and Finnish languages. When compared to a recent paper (Shahnawazuddin et al., 2020) for PF-STAR verification set, our system gives a 50 % relative improvement. Furthermore, when we run the adult test sets VC_E and VC_H again using the final system trained with shared system with mixed of adults' and children's data, we found out that the performance remains the same. This means that we can now use the same model for recognizing both adults and children.

6 Conclusion

We have presented a speaker verification system based on a shared neural embedding space for adults and children. The neural network consists of a cascade of LSTM layers and a NetVLAD aggregation layer, and uses the AM-softmax loss in training. We have demonstrated that the system achieves promising results with adults and children. Because the child data is limited, we trained a shared system with mixed adult and child data to capture more acoustic variability. The shared sys-

tem gives a 54% and 43% relative improvement for children compared to the separate children’s and adult systems. For adults, the shared system gives the same performance as compared to the adult system. Finally, we can conclude that this shared system can be now used for both children and adults.

7 Acknowledgment

This work was supported by the Academy of Finland (grant 329267) and EU’s Horizon 2020 research and innovation programme via the project MeMAD (GA 780069).

References

- Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307.
- A. Batliner, M. Blomberg, S. D’Arcy, D. Elenius, D. Giuliani, M. Gerosa, C. Hacker, M. Russell, and M. Wong. 2005. The PF.STAR children’s speech corpus. In *Proc. INTERSPEECH*, pages 2761–2764.
- Hervé Bredin. 2017. Tristounet: triplet loss for speaker turn embedding. In *Proc. ICASSP*, pages 5430–5434. IEEE.
- J. S. Chung, A. Nagrani, and A. Zisserman. 2018. Voxceleb2: Deep speaker recognition. In *Proc. INTERSPEECH*.
- Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. 2010. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798.
- Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699.
- Rosa González Hautamäki, Ville Hautamäki, and Tomi Kinnunen. 2019. On the limits of automatic speaker verification: Explaining degraded recognizer scores through acoustic changes resulting from voice disguise. *The Journal of the Acoustical Society of America*, 146(1):693–704.
- Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. 2016. End-to-end text-dependent speaker verification. In *Proc. ICASSP*, pages 5115–5119. IEEE.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Sergey Ioffe. 2006. Probabilistic linear discriminant analysis. In *European Conference on Computer Vision*, pages 531–542. Springer.
- Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456.
- T. Kaseva, A. Rouhe, and M. Kurimo. 2019. Spherediar: An effective speaker diarization system for meeting data. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 373–380.
- Ronald Allan Cole Khaldoun Shobaki, John-Paul Hosom. 2007. CSLU: Kids’ Speech Version 1.1 LDC2007S18. Web Download. Philadelphia. In *Linguistic Data Consortium*.
- H. Kumar Kathania, Sudarsana Reddy Kadiri, P. Alku, and M. Kurimo. 2020. Study of formant modification for children asr. In *Proc. ICASSP*, pages 7429–7433.
- Sungbok Lee, Alexandros Potamianos, and Shrikanth Narayanan. 1999. Acoustics of children’s speech: Developmental changes of temporal and spectral parameters. *The Journal of the Acoustical Society of America*, 105(3):1455–1468.
- Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. 2017. Deep speaker: an end-to-end neural speaker embedding system. *arXiv preprint arXiv:1705.02304*.
- Yutian Li, Feng Gao, Zhijian Ou, and Jiasong Sun. 2018. Angular softmax loss for end-to-end speaker verification. In *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 190–194. IEEE.
- Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. 2017. Spheredface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220.
- Yi Liu, Liang He, and Jia Liu. 2019. Large margin softmax loss for speaker verification. *arXiv preprint arXiv:1904.03479*.
- Mitchell McLaren, Luciana Ferrer, Diego Castan, and Aaron Lawson. 2016. The speakers in the wild (SITW) speaker recognition database. In *Proc. INTERSPEECH*, pages 818–822.
- A. Nagrani, J. S. Chung, and A. Zisserman. 2017. Voxceleb: a large-scale speaker identification dataset. In *Proc. INTERSPEECH*.
- Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda. 2018. Attentive statistics pooling for deep speaker embedding. *arXiv preprint arXiv:1803.10963*.

- Mirco Ravanelli and Yoshua Bengio. 2018. Speaker recognition from raw waveform with sincnet. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028. IEEE.
- Antti Rosti, Anssi Ramo, Teemu Saarelainen, and Jari Yli-Hietanen. 1998. Speechdat finnish database for the fixed telephone network. *Tech. Rep., Tampere University of Technology*.
- Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.
- Saeid Safavi, Maryam Najafian, Abualsoud Hanani, Martin Russell, Peter Jancovic, and Michael Carey. 2012. Speaker recognition for children’s speech. In *Proc. INTERSPEECH*, volume 3.
- Saeid Safavi, Martin Russell, and Peter Jančovič. 2018. Automatic speaker, age-group and gender identification from children’s speech. *Computer Speech Language*, 50:141 – 156.
- S. Shahnawazuddin, Nagaraj Adiga, B Tarun Sai, Waquar Ahmad, and Hemant K. Kathania. 2019. Developing speaker independent asr system using limited data through prosody modification based on fuzzy classification of spectral bins. *Digital Signal Processing*, 93:34 – 42.
- S. Shahnawazuddin, W. Ahmad, N. Adiga, and A. Kumar. 2020. In-domain and out-of-domain data augmentation to improve children’s speaker verification system in limited data scenario. In *Proc. ICASSP*, pages 7554–7558.
- David Snyder, Daniel Garcia-Romero, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur. 2019. Speaker recognition for multi-speaker conversations using x-vectors. In *Proc. ICASSP*, pages 5796–5800. IEEE.
- David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. 2018. X-vectors: Robust dnn embeddings for speaker recognition. In *Proc. ICASSP*, pages 5329–5333. IEEE.
- Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. 2018. Generalized end-to-end loss for speaker verification. In *Proc. ICASSP*, pages 4879–4883. IEEE.
- Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. 2018a. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930.
- Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. 2018b. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274.
- Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopez Moreno. 2018c. Speaker diarization with lstm. In *Proc. ICASSP*, pages 5239–5243. IEEE.
- Guillaume Wisniewski, Hervé Bredin, Grégory Gelly, and Claude Barras. 2017. Combining speaker turn embedding and incremental structure prediction for low-latency speaker diarization. In *Proc. INTERSPEECH*.
- Xu Xiang, Shuai Wang, Houjun Huang, Yanmin Qian, and Kai Yu. 2019. Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition. *arXiv preprint arXiv:1906.07317*.
- Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Senior. 2019. Utterance-level aggregation for speaker recognition in the wild. In *Proc. ICASSP*, pages 5791–5795. IEEE.
- Hossein Zeinali, Shuai Wang, Anna Silnova, Pavel Matějka, and Oldřich Plchot. 2019. But system description to voxceleb speaker recognition challenge 2019.
- Aonan Zhang, Quan Wang, Zhenyao Zhu, John Paisley, and Chong Wang. 2019. Fully supervised speaker diarization. In *Proc. ICASSP*, pages 6301–6305. IEEE.
- Tianyan Zhou, yong zhao, Jinyu Li, Yifan Gong, and Jian Wu. 2019. Cnn with phonetic attention for text-independent speaker verification. In *Automatic Speech Recognition and Understanding Workshop*. IEEE.

Spectral modification for recognition of children’s speech under mismatched conditions

Hemant Kathania, Sudarsana Reddy Kadiri, Paavo Alku and Mikko Kurimo

Department of Signal Processing and Acoustics, Aalto University, Finland

(hemant.kathania, sudarsana.kadiri, paavo.alku, and mikko.kurimo)@aalto.fi

Abstract

In this paper, we propose spectral modification by sharpening formants and by reducing the spectral tilt to recognize children’s speech by automatic speech recognition (ASR) systems developed using adult speech. In this type of mismatched condition, the ASR performance is degraded due to the acoustic and linguistic mismatch in the attributes between children and adult speakers. The proposed method is used to improve the speech intelligibility to enhance the children’s speech recognition using an acoustic model trained on adult speech. In the experiments, WSJCAM0 and PFSTAR are used as databases for adults’ and children’s speech, respectively. The proposed technique gives a significant improvement in the context of the DNN-HMM-based ASR. Furthermore, we validate the robustness of the technique by showing that it performs well also in mismatched noise conditions.

Index Terms: Children speech recognition, Spectral sharpening, Spectral tilt, DNN.

1 Introduction

Recent advances in ASR have impacted many applications in various fields, such as education, entertainment, home automation, and medical assistance (Vajpai and Bora, 2016). These applications can benefit children in their daily life, in playing games, reading tutors (Mostow, 2012), and learning both native and foreign languages (Evanini and Wang, 2013; Yeung and Alwan, 2019).

The task of speech parameterization for the front-end aims at a compact representation that captures the relevant information in the speech signal by using short-time feature vectors. The two

commonly used feature sets are Mel-frequency cepstral coefficients (MFCC) (Davis and Mermelstein, 1980) and the perceptual linear prediction cepstral coefficients (PLPCC) (Lee et al., 1999; Huber et al., 1999). Speech of adults and children have large acoustic and linguistic differences (Lee et al., 1999; Narayanan and Potamianos, 2002; Potaminaos and Narayanan, 2003; Gerosa et al., 2009). Both the Mel-filterbank and PLP coefficients are better suited for adults as they provide better resolution for low-frequency contents while a greater degree of averaging happens in the high-frequency range (Davis and Mermelstein, 1980; Hermansky, 1990a).

In the case of children’s speech, more relevant information is available in the high-frequency range. Therefore, to enhance the system performance, a better resolution needs to be used for the high-frequency range. Previous studies have also shown that formant sharpening is helpful for increasing speech intelligibility (Chennupati et al., 2019; Zorila Tudor-Catalin and Yannis, 2012; Potaminaos and Narayanan, 2003; Kathania et al., 2014). Motivated by these observations, we suggest to modify the speech spectrum by formant sharpening and spectral tilt reduction.

In (Potamianos and Narayanan, 2003; Kathania et al., 2014, 2016), it was shown that the word error rate (WER) in recognition of children’s speech is much higher than that of adult speech and specifically under mismatched and noisy conditions. The problems are due to higher inter-speaker variance caused by the development of the vocal tract, leading to different formant locations and spectral distribution (Hermansky, 1990b), and due to the inaccuracy in pronunciation and grammar caused by language acquisition. Most importantly, the insufficient training data limits the performance because collecting large speech databases of children’s speech is hard. Adult speech corpora normally contain hun-

dreds or thousands of hours of data, while most publicly available corpora for children’s speech have less than 100 hours of data (Panayotov et al., 2015; Claus et al., 2013). Therefore, it is necessary that ASR systems built for children are robust for various mismatched conditions.

In this paper, a spectral sharpening and tilt reduction method is proposed to enhance the intelligibility of children’s speech to boost the ASR system performance under mismatched conditions. Spectral sharpening and spectral tilt reduction have been used in enhancement of speech intelligibility in noise (Chennupati et al., 2019; Zorila Tudor-Catalin and Yannis, 2012). In this study, it is shown that the MFCC and PLPCC features computed after the spectral modification (referred to as SS-MFCC and SS-PLPCC) are found to outperform the conventional MFCC and PLPCC features. This is demonstrated by both the spectral analyses and experimental evaluations in this paper. The robustness of the technique is further validated by showing that it performs well in mismatched noise conditions also.

The remaining of this paper is presented as follows: In Section 2, the proposed spectral sharpening and tilt reduction technique is discussed. In Section 3, the speech corpora and ASR specifications are described. The results of the proposed method are presented in Section 4. In Section 5, the effects of noisy environment on the proposed method are discussed. Finally, the paper is concluded in Section 6.

2 The spectral modification method

The proposed spectral modification technique consists of formant sharpening and spectral tilt reduction as described below and depicted in the block diagram in Fig 1. From the spectral examples shown in Fig 2 and spectrograms shown in Fig 3, we can observe that the proposed method enhances formant peaks and the level of higher frequencies.

2.1 Adaptive spectral sharpening

The formant information is important for recognizing speech, and Adaptive Spectral Sharpening (ASS) is a method that emphasizes the formant information (Zorila Tudor-Catalin and Yannis, 2012). For sharpening of formants, an approach that was motivated in speech intelligibility is utilised (Zorila Tudor-Catalin and Yannis, 2012). In this method, the magnitude spectrum is

extracted using the SEEVOC method (Paul, 1981) for the pre-emphasized voice speech frame. The adaptive spectral sharpening at frame t is given by

$$H_s(\omega, t) = \left(\frac{E(\omega, t)}{T(\omega, t)} \right)^\beta, \quad (1)$$

where $E(\omega, t)$ is the estimated spectral envelope computed using the SEEVOC method and $T(\omega, t)$ is the spectral tilt for frame t . Spectral tilt $T(\omega, t)$ is computed using cepstrum and is given by

$$\log T(\omega) = C_0 + 2C_1 \cos(\omega). \quad (2)$$

Here C_m is the m th cepstral coefficients and is given by

$$C_m = \frac{1}{\left(\frac{N}{2} + 1\right)} \sum_{k=0}^{\frac{N}{2}} E(\omega_k) \cos(m\omega_k). \quad (3)$$

Formant sharpening is performed using Eq. (1) by varying β . Typically, the value of β is higher for low signal-to-noise ratio (SNR) values and lower for high SNR values. In this study, we have investigated the extent of spectral sharpening by varying the β parameter from 0.15 to 0.35. Note that spectral sharpening is performed only in voiced segments using probability of voicing as defined in (Zorila Tudor-Catalin and Yannis, 2012).

2.2 Spectral tilt modification

Apart from spectral sharpening, we also perform fixed spectral tilt modification ($H_r(\omega)$) to boost the region between 1 kHz and 4 kHz by 12 dB and to reduce the level of frequencies below 500 Hz (by 6 dB/octave). The resulting magnitude spectrum for a frame after the ASS and fixed spectrum tilt modification is given by

$$\hat{E}(\omega) = E(\omega)H_s(\omega)H_r(\omega) \quad (4)$$

The modified magnitude spectrum ($\hat{E}(\omega)$) is combined with the original phase spectrum for reconstructing the signal using IDFT and Overlap-and-Add (OLA) (Rabiner and Gold, 1975).

A schematic block diagram describing the steps involved in the proposed method is shown in Fig 1. Fig 2 illustrates the effect of spectral modification for a voiced child’s speech segment. Here the blue curve is the spectrum of the original speech segment and the red curve is the modified speech spectrum. From the figure, it can be seen that

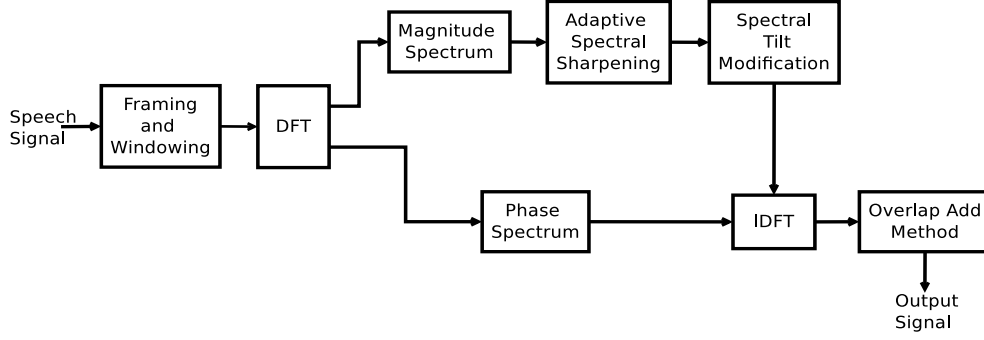


Figure 1: Block diagram of the spectral modification method.

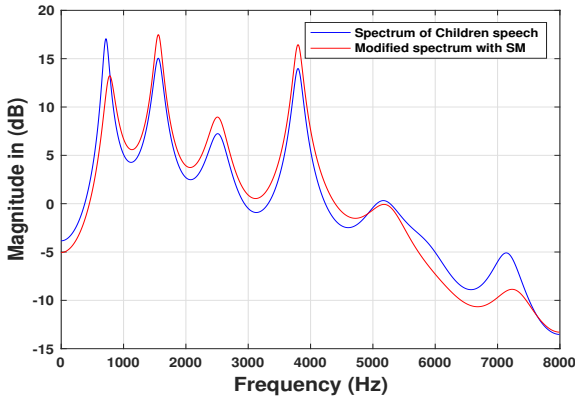


Figure 2: Spectrum for a segment of child's speech (blue) and the corresponding spectrum after the spectral modification (SM) (red).

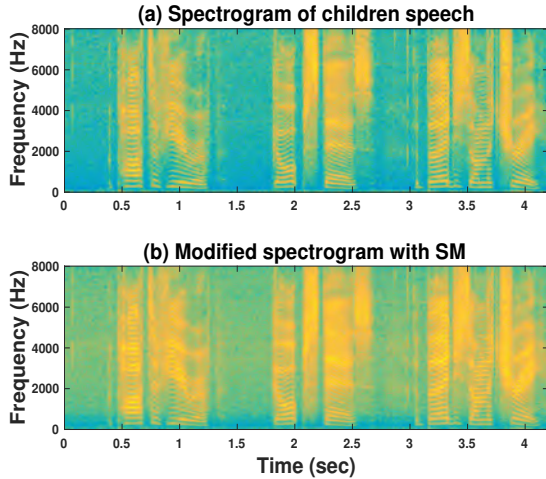


Figure 3: Spectrogram for a segment of child's speech shown in (a), and the corresponding spectrogram after spectral modification shown in (b).

formants are sharpened by the proposed method (red curve). Specifically, it can be clearly seen that formants are more prominent in the region of 1 kHz to 4 kHz for the proposed method (red

curve), which is due to the spectral modification as described in Section 2.2. Furthermore, illustrations of the spectrograms are shown in Fig 3. Fig 3 (a) shows the child's original spectrogram before modifications and Fig 3 (b) shows the corresponding spectrogram after the proposed spectral modification (SM) method. Again it can be observed from Fig 3(b) that the spectrogram has a larger high-frequency emphasis compared to spectrogram in Fig 3(a), due to spectral modification in the proposed method.

3 Data and Experimental setup

This section describes the speech corpora (adult and children), front-end speech features and specifications of ASR system.

3.1 Speech Corpora

Adult speech data used in this work was obtained from WSJCAM0 (Robinson et al., 1995). Children's speech data was obtained from the PF-STAR corpus (Batliner et al., 2005) to simulate a mismatched ASR task. Both the WSJCAM0 and PF-STAR corpora are British English speech databases. Details of both corpora are given in Table 1

3.2 Front-end speech parameterization

The speech data was first pre-emphasized with a first order FIR high-pass filter (with zero at $z = 0.97$). For frame-blocking, overlapping Hamming windows with a length of 20 ms and an overlap of 50% were used. 13-dimensional MFCCs were extracted using 40 channels. The 13-dimensional base MFCC features were then spliced in time taking a context size of 9 frames. Time-splicing resulted in 117-dimensional features vectors. Linear discriminant analysis (LDA) and maximum-likelihood linear transformation (MLLT) were

Table 1: Speech corpora details for WSJCAM0 and PFSTAR used in ASR

Corpus	WSJCAM0		PF-STAR	
Language	British English		British English	
Purpose	Training	Testing	Training	Testing
Speaker group	Adult	Adult	Child	Child
No. of speakers	92	20	122	60
Speaker age	> 18 years	> 18 years	4-14 years	4-13 years
No. of words	132,778	5,608	46974	5067
Duration (hrs.)	15.50	0.60	8.3	1.1

used to reduce the feature vector dimension from 117 to 40. The 13-dimensional base PLPCC features were derived using 12th-order linear prediction (LP) analysis. Cepstral mean and variance normalization (CMVN) as well as feature-space maximum-likelihood linear regression (fM-LLR) were performed next to enhance robustness with respect to speaker-dependent variations. The required fM-LLR transformations for the training and test data were generated through speaker adaptive training.

The MFCC and PLPCC features computed after the proposed spectral modification (i.e., spectral sharpening and tilting) are referred to as SS-MFCC and SS-PLPCC, respectively. ASR results are given for the baseline features (MFCC and PLPCC) and the proposed features (SS-MFCC and SS-PLPCC) for all the experiments conducted in this paper.

3.3 ASR system specifications

To build the ASR system on the adult speech data from the WSJCAM0 speech corpus, the Kaldi toolkit (Povey et al., 2011) was used. Context-dependent hidden Markov models (HMM) were used for modeling the cross-word triphones. Decision tree-based state tying was performed with the maximum number of tied-states (senones) being fixed at 2000. A deep neural network (DNN) was used in acoustic modeling. Prior to learning parameters of the DNN-HMM-based ASR system, the fM-LLR-normalized feature vectors were time-spliced once again considering a context size of 9 frames. The number of hidden layers in the DNN was set to 5 with 1024 hidden nodes in each layer. The nonlinearity in the hidden layers was modeled

using the *tanh* function. The initial learning rate for training the DNN-HMM parameters was set at 0.005 which was reduced to 0.0005 in 15 epochs. The minibatch size for neural net training was set to 512.

For decoding the test set for adults, the MIT-Lincoln 5k vocabulary Wall Street Journal bi-gram language model (LM) was used. The perplexity of this LM for the adult test set is 95.3 while there are no out-of-vocabulary (OOV) words. Furthermore, a lexicon consisting of 5850 words including pronunciation variants was used. While decoding the test set for children’s speech, a 1.5k domain-specific bigram LM was used. This bigram LM was trained on the transcripts of speech data in PF-STAR after excluding those corresponding to the test set of children’s speech. The domain-specific LM has an OOV rate of 1.20% and perplexity of 95.8 for the test set of children’s speech. In total 1969 words used including pronunciation variations in lexicon for decoding the children’s test set.

4 Results and discussion

The baseline WERs for children’s test set in the DNN-HMM systems is 19.76% and 20.00% for the MFCC and PLPCC acoustic features respectively (see Table 2). In order to improve the recognition performance, the spectral sharpening technique is applied to mitigate the spectral differences between adults’ and children’s speech. The spectral sharpening algorithm includes the tunable β parameter according to Eq. (1), and this parameter was varied from 0.15 to 0.35 to sharpen the spectral peaks (formants). The WERs obtained with varying sharpening parameter are shown in Figure 4. From the figure, it can be observed that the best WER was obtained with $\beta = 0.25$. The remaining experiments are carried out using this value of β .

The baseline WERs for children’s test set with respect to the DNN-HMM-based ASR systems trained using the MFCC and PLPCC features are given in Table 2. The MFCC and PLPCC features computed after the formant modification are denoted as SS-MFCC and SS-PLPCC, respectively in Table 2. A notable reduction in WER can be observed for both the features.

For further analysis, the children test data was divided into three different test sets based on age groups: 4 – 6 years, 7 – 9 years, and 10 – 13 years. Table 3 shows the results for baseline and

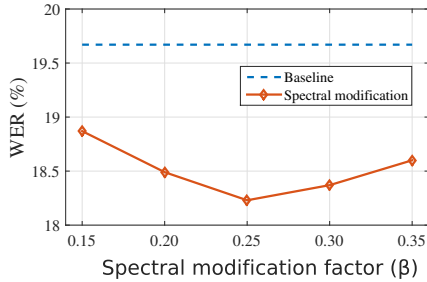


Figure 4: WER results depicting the effect of spectral modification (for varying the β parameter) on recognition of children’s speech using an DNN-HMM system trained using adult speech.

proposed features for three age groups. It can be seen that the proposed approach improves the results in all the age groups for both of the proposed features, SS-MFCC and SS-PLPCC. We have also conducted significance test and notice that signed pair comparison found significant difference between the two approaches at level $p < 0.01$.

To further validate the effectiveness of the proposed modification method, another DNN-HMM-based ASR system was developed by pooling together speech data from training sets of both adults and children. For children’s speech, the training set derived from PF-STAR consisted of 8.3 hours of speech by 122 speakers. The total number of utterances in this training set was equal to 856 with a total of 46974 words. The training set of adult speakers consisted of 15.5 hours of speech from 92 speakers (both male and female). Further, the training set comprised 132,778 words and the total number of utterances was 7852. The developed ASR system exhibits a lower degree of acoustic/linguistic mismatch due to the pooling of children’s speech into training. As a result, the baseline WERs for the developed system (given in Table 2) are significantly lower when compared to those obtained with respect to the ones trained on adult speech only. Still, further reductions in WERs are achieved when the spectral modification technique is applied to enhance the speech intelligibility as shown in Table 2.

5 Experiments in Noisy conditions

To further validate the proposed technique, noise robustness of the spectral modification technique was studied. Four different noises (babble, white, factory and volvo noise) extracted from NOISEX-92 (Varga and Steeneken, 1993) were added to the

Table 2: WERs of the baseline and proposed spectral modification method for children’s ASR. The performance evaluation is done separately using two ASR systems: a system trained with only adult speech from WSJCAM0 and a system trained by pooling also children’s speech.

Training Data	Testing Data	WER in (%)			
		DNN-HMM (Acoustic Model)			
		PLPCC	SS-PLPCC	MFCC	SS-MFCC
Adult speech	Children’s speech	20.00	19.38	19.76	18.23
Adult + children’s speech	Children’s speech	12.89	12.43	12.26	11.70

Table 3: WERs for the age-wise grouped children speech test sets with respect to adults data trained ASR systems demonstrating the effect of the proposed spectral modification.

Age wise setup	WER (in %)			
	PLPCC	SS-PLPCC	MFCC	SS-MFCC
4 - 6	72.36	70.18	70.48	68.18
7 - 9	20.11	17.24	19.38	16.20
10 - 13	12.35	11.72	11.78	10.53

test data under varying SNR levels. The noisy test sets were then decoded using the acoustic models trained with clean speech. WERs in the case of adult/child mismatched testing are given in Table 4 for SNR values of 5 dB, 10 dB, and 15 dB. While the MFCC features seem slightly more robust to additive noise than the PLPCC features, the spectral modification reduces WER clearly for both of the acoustic features (denoted as SS-MFCC and SS-PLPCC) at the three different SNR levels. Hence, it can be concluded that the spectral sharpening of formant peaks improves the ASR performance also in various noisy conditions.

6 Conclusion

This work explores spectral modification (sharpening of formants and reduction of spectral tilt) to achieve robust recognition of children’s speech under mismatched conditions. The explored spectral modification technique is observed to enhance ASR of children’s speech for both the MFCC and PLPCC features. Also, ASR results are analyzed for different age-groups and it was found that for all the age-groups there exists an improvement

Table 4: WERs of the proposed spectral modification method for children’s speech test set under varying additive noise conditions.

Noise Type	SNR (dB)	WER in (%)			
		PLPCC	SS-PLPCC	MFCC	SS-MFCC
Babble	5dB	83.69	82.67	79.70	80.35
	10dB	64.62	58.36	59.7	56.41
	15dB	48.47	42.61	40.34	38.08
White	5dB	86.54	83.61	87.40	86.25
	10dB	79.01	77.26	73.78	72.62
	15dB	66.79	63.58	54.00	53.46
Factory	5dB	86.54	83.61	92.32	90.86
	10dB	67.13	65.96	68.96	66.95
	15dB	49.32	48.65	45.33	43.55
Volvo	5dB	34.71	26.22	26.12	24.70
	10dB	29.16	24.58	23.10	22.03
	15dB	25.61	22.89	21.64	20.75

with the proposed approach compared to baseline. Further, improvements were also observed in mismatch conditions caused by additive noise.

7 Acknowledgements

This work was supported by the Academy of Finland (grant 329267). The computational resources were provided by Aalto ScienceIT.

References

A. Batliner, M. Blomberg, S. D’Arcy, D. Elenius, D. Giuliani, M. Gerosa, C. Hacker, M. Russell, and M. Wong. 2005. The PF.STAR children’s speech corpus. In *Proc. INTERSPEECH*, pages 2761–2764.

Nivedita Chennupati, Sudarsana Reddy Kadiri, and B. Yegnanarayana. 2019. Spectral and temporal manipulations of sff envelopes for enhancement of speech intelligibility in. *Computer Speech Language*, 54:86 – 105.

Felix Claus, Hamurabi Gamboa-Rosales, Rico Petrick, Horst-Udo Hain, and Rüdiger Hoffmann. 2013. A survey about databases of children’s speech. In *14th Annual Conference of the International Speech Communication Association At: Lyon, France*, pages 2410–2414.

S. Davis and P. Mermelstein. 1980. <https://doi.org/10.1109/TASSP.1980.1163420> Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 28(4):357–366.

K. Evanini and X. Wang. 2013. Automated speech scoring for non-native middle school students with multiple task types. In *Proc. INTERSPEECH*, pages 2435–2439.

Matteo Gerosa, Diego Giuliani, Shrikanth Narayanan, and Alexandros Potamianos. 2009. A review of ASR technologies for children’s speech. In *Proc. Workshop on Child, Computer and Interaction*.

H. Hermansky. 1990a. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 57(4):1738–52.

Hynek Hermansky. 1990b. Perceptual linear predictive (plp) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4):1738–1752.

Jessica Huber, Elaine Stathopoulos, Gina Curione, Theresa Ash, and Kenneth Johnson. 1999. <https://doi.org/10.1121/1.427150> Formants of children, women, and men: The effects of vocal intensity variation. *The Journal of the Acoustical Society of America*, 106:1532–42.

H. K. Kathania, S. Shahnawazuddin, G. Pradhan, and A. B. Samaddar. 2016. Experiments on children’s speech recognition under acoustically mismatched conditions. In *2016 IEEE Region 10 Conference (TENCON)*, pages 3014–3017.

H. K. Kathania, S. Shahnawazuddin, and R. Sinha. 2014. Exploring hlda based transformation for reducing acoustic mismatch in context of children speech recognition. In *2014 International Conference on Signal Processing and Communications (SPCOM)*, pages 1–5.

Sungbok Lee, Alexandros Potamianos, and Shrikanth S. Narayanan. 1999. Acoustics of children’s speech: Developmental changes of temporal and spectral parameters. *The Journal of the Acoustical Society of America*, 105(3):1455–1468.

J. Mostow. 2012. Why and how our automated reading tutor listens. In *Proc. INTERSPEECH*, 4.

S. Narayanan and A. Potamianos. 2002. Creating conversational interfaces for children. *IEEE Transactions on Speech and Audio Processing*, 10(2):65–78.

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

D Paul. 1981. The spectral envelope estimation vocoder. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(4):786–794.

A. Potamianos and S. Narayanan. 2003. Robust recognition of children’s speech. *IEEE Transactions on Speech and Audio Processing*, 11(6):603–616.

- A. Potaminaos and S. Narayanan. 2003. Robust Recognition of Children Speech. *IEEE Transactions on Speech and Audio Processing*, 11(6):603–616.
- Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The kaldi speech recognition toolkit. In *Proc. ASRU*.
- Lawrence R Rabiner and Bernard Gold. 1975. Theory and application of digital signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc.*
- T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals. 1995. WSJCAM0: A British English speech corpus for large vocabulary continuous speech recognition. In *Proc. ICASSP*, volume 1, pages 81–84.
- J. Vajpai and A. Bora. 2016. Industrial applications of automatic speech recognition. *International Journal of Engineering Research and Applications*, 6(3):88–95.
- Andrew Varga and Herman J.M. Steeneken. 1993. Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Communication*, 12(3):247–251.
- Gary Yeung and Abeer Alwan. 2019. A Frequency Normalization Technique for Kindergarten Speech Recognition Inspired by the Role of fo in Vowel Perception. In *Proc. INTERSPEECH*, pages 6–10.
- Kandia Varvara Zorila Tudor-Catalin and Stylianos Yannis. 2012. Speech-in-noise intelligibility improvement based on spectral shaping and dynamic range compression. *INTERSPEECH*, pages 635 – 638.

A Baseline Document Planning Method for Automated Journalism

Leo Leppänen

University of Helsinki
Department of Computer Science
leo.leppanen@helsinki.fi

Hannu Toivonen

University of Helsinki
Department of Computer Science
hannu.toivonen@helsinki.fi

Abstract

In this work, we present a method for content selection and document planning for automated news and report generation from structured statistical data such as that offered by the European Union’s statistical agency, Eurostat. The method is driven by the data and is highly topic-independent within the statistical dataset domain. As our approach is not based on machine learning, it is suitable for introducing news automation to the wide variety of domains where no training data is available. As such, it is suitable as a low-cost (in terms of implementation effort) baseline for document structuring prior to introduction of domain-specific knowledge.

1 Introduction

Automated generation of news texts from structured data – often referred to as ‘automated journalism’ (Graefe, 2016; Dörr, 2015; Caswell and Dörr, 2018) or ‘news automation’ (Linden, 2017; Sirén-Heikel et al., 2019; Dierickx, 2019) – is of great interest to various news producers. It is seen as a way of ‘providing efficiency, increasing output and aiding in reallocating resources to pursue quality journalism’ (Sirén-Heikel et al., 2019, p. 47). While data-to-text NLG systems are still far from common especially among the smaller, regional news industry players, at least among the larger newsrooms the use of NLG approaches has clearly been established (Fanta, 2017).

While secrecy in the industry makes it difficult to establish the commercial reality as an outsider, the limited available evidence indicates that commercial automated journalism is mostly done using rule-based methods despite a surge of academic interest in increasingly complex neural methods for NLG (e.g. Puduppully et al., 2019; Ferreira et al.,

2019): Interviews of news automation users indicate that the employed methods are mostly based on templates (Sirén-Heikel et al., 2019), as are the few open source code repositories of real-world news automation systems (Yleisradio, 2018). Indeed, some NLG industry experts believe that especially end-to-end neural models do not match customer needs at this time (Reiter, 2019).

Contributing factors include a lack of control (Reiter, 2019); issues with hallucination of non-grounded output (Nie et al., 2019; Dušek et al., 2019; Reiter, 2018); the difficulty in surgically correcting any issues identified in trained neural models beyond additional training; as well as the difficulty of establishing what the ‘worst case’ performance of a neural model is.

In addition, we believe that that while neural NLG methods are theoretically highly transferable, the *practical* transferability of neural NLG solutions to many news domains is limited by a lack of training data. While newsrooms have extensive archives of news text, these are rarely associated with the matching data that is the ‘input’ for each piece of news text (E.g., MacKová and Sido, 2020, pp. 43–44, Kanerva et al., 2019, p. 247). At the same time, the non-trainable methods for NLG, too, suffer from difficulties in transferability and reusability (Linden, 2017).

In this work, we investigate document planning (selecting what content and in what order should appear in the document) for structured, statistical data-to-text NLG in the context of automated journalism targeting human journalists. We are not in search of a perfect method, but rather something that is relatively easy to implement as a subdomain-independent baseline and which can then be enhanced with domain-specific processing later-on. Such a method would make it easier to introduce automated journalism solutions to completely new subdomains within the larger statistical data domain.

2 Structuring Hard News

When queried for insight into news structure, journalists and academics often recite the concept of the “(inverted) news pyramid”, where the news article is structured so that the order in which information appears in the text reflects the journalist’s belief about the importance of the piece of information (Thomson et al., 2008). While the precise origin of the structure is not clear (Pöttker, 2003), it has become so prototypical that it is held self-evident in the journalistic trade literature: “*Every journalist knows how to write a traditional news text: start with the most important thing and continue until you have either said everything relevant or the space reserved for the story runs out*” (Sulopuisto, 2018, translated from Finnish).

A more rigorous analysis of the structures employed in ‘hard’ news is presented by White (1997), who argues that hard news articles have an ‘orbital’ structure consisting of a *nucleus* which represents the main point of the article and *satellites* that give context and additional information about the nucleus. White (1997) assigns the role of the nucleus to the combination of the headline and the lead paragraph of the article, and describes the subsequent paragraphs as the satellites. White (1997) identifies five possible relations between a satellite and the nucleus: elaboration, cause-and-effect, justification, contextualization and appraisal. Thomson et al. (2008), in turn, identify that the satellites can elaborate, reiterate, describe causes or consequences, contextualize or provide additional assessment. An important observation is that – as indicated by ‘orbital’ – these satellites are relatively freely reorderable without affecting readability or meaning. Together, these two observations indicate that a good document plan for hard news (1) prioritizes more newsworthy items and (2) contains some overarching theme (exemplified by the nucleus) so that the text as a whole is coherent, i.e. the satellites are in some way related to the nucleus.

The relations identified by White (1997) and Thomson et al. (2008) are highly similar to those identified in the more general Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), which uses similar nucleus-satellite terminology. However, whereas White (1997) and Thomson et al. (2008) analyze news text on the level of paragraphs, RST can be applied on a more fine-grained level to much shorter text spans. As RST shows that similar relations can be applied on a sub-paragraph

level, we hypothesize that a reasonably approximation of a news article might be constructed by applying White’s (1997) orbital theory also *within* paragraphs, by considering the first *sentence* of the paragraph a nucleus, and the others as satellites.

Importantly, we interpret the orbital theory of news structuring to suggest that – as the satellites are freely orderable – the actual *type* of relation is not as important for document planning as knowing that *some* relation exists between the satellite and the nucleus. We hypothesize that while identifying whether a specific (RST) relation exists between two arbitrary pieces of information requires domain knowledge, an approximation of whether two arbitrary pieces of information are related in *some* way could be obtained by inspecting their similarity in a domain-independent fashion.

That is, we expect that a piece of information regarding the US health care funding in 2020 is more likely to be related in *some way* to a piece of information discussing the US health care funding in 2020 than to another piece of information discussing the health care funding in Sweden in 1978. If a heuristic or similarity measure identifying such relations could be identified, it could be used together with some estimate of newsworthiness to construct paragraph and document plans that seek to maximize both the key aspects identified above: newsworthiness and the relatedness of the content.

As noted in the introduction, there is a distinction between the theoretical and the practical transferability of neural processing methods. We believe that a good baseline document planning and content selection approach should avoid the need for training data present in the many of recently proposed document planning and content selection approaches. This rules out as unsuitable most recent work that are based on learning from an aligned corpus of data and human-written texts, such as Angeli et al. (2010), Konstas and Lapata (2013), Wiseman et al. (2017), Zhang et al. (2017), Li and Wan (2018), Dou et al. (2018) and Puduppully et al. (2019).

Outside of these trainable approaches, to our knowledge, most other document planning approaches are based on ‘*hand-engineered*’ (Konstas and Lapata, 2013), domain-specific methods. A highly relevant survey of various document planning methods is presented by Gkatzia (2016). While these previous works are – to at least some degree – domain-specific, they establish concepts

and ideas that are highly relevant for our goal. Both Hallett et al. (2006) and Gatt et al. (2009) describe a core set of information, called ‘summary spine’ or ‘key events’, that they hold as more important than the rest of the available information. They, as well as Banaee et al. (2013), also employ a numeric estimate of importance. Demir et al. (2010) identify that content already selected for inclusion in the document plan affects how well suited so-far unselected content is for inclusion. Sripada et al. (2003) identify Gricean maxims (Grice, 1975) as providing requirements for document planning and content selection.

3 Context

Our work on document planning is done in the context of a series of data-to-text NLG applications producing short highlights of structured statistical data. Importantly, the applications are intended to be deployed in contexts where they must be able to produce texts highlighting between 10 and 30 data points from datasets measured in 100,000s of data points. The resulting texts are intended to both alert journalists to potential news and to provide them with a starting place from which to write the final news text.

Our system, adapted from Leppänen et al. (2017a), is based on a pipeline of components with dedicated responsibilities similar to those described by Reiter and Dale (2000) and Reiter (2007). For this work, the relevant part of the architecture is the Document Planner component. This component receives as input two sets of *message* data structures, an example of which is shown in Table 1.¹ The messages are extracted automatically from tables of statistical data obtained from Eurostat.

The *core set* contains messages that are known to be highly relevant to the generation task. Unlike the ‘summary spine’ of Hallett et al. (2006), the set is unlinked and unordered, and not all members of the set are guaranteed to be included in the document plan. The *expanded set*, contains messages that *can* be, but are not guaranteed to be, relevant for the document. Expressed using the terminology from Section 2, we assume that only messages in the core set can be nuclei, while messages from either set can be satellites.

These core and expanded sets are determined automatically from user input. When requesting

¹The concrete implementation details are somewhat more complex. We omit details irrelevant for this work.

a new text, the user of the system must define a dataset the text is to be generated from, for example the consumer price data available from Eurostat. This dataset is then divided into the core set and the expanded set by the user when they select what country the generated text should focus on. For example, if the user were to select that the text should discuss French consumer prices, the core set would contain all data from the consumer price dataset that pertains directly to France, while the rest of the consumer price dataset (including data pertaining to the UK, Finland, Croatia, etc.) would be set as the expanded set.

We estimate each message’s ‘newsworthiness’ using the Interquartile Range based method described by Leppänen et al. (2017b) with the values scaled to have mean 0 and standard deviation 1 for the purposes of this computation. The resulting value is conceptually similar to ‘importance’ of Gatt et al. (2009) and ‘risk’ of Banaee et al. (2013). The IQR based method compares each data point in turn to a larger distribution, giving it higher scores the further it is from the area between the first and the third quartile of the larger distribution. Values between the quartiles are given a minimal, uniform, score that is dependent on the shape of the distribution. In other words, higher IQR values indicate that the value is more of an outlier compared to the rest of related data in the dataset. As such, it captures a degree of ‘unexpectedness’, which is an important aspect of newsworthiness (Galtung and Ruge, 1965).

We do not use the domain-specific parts of the method described by Leppänen et al. (2017b). That is, we make no value judgement of whether messages pertaining to French consumer prices are more newsworthy than messages pertaining to Croatian consumer prices, nor do we make judgements of whether changes in the price of education are more or less newsworthy than changes in the price of alcohol and tobacco. However, we do weight the scores so that messages with the `timestamp` field being closer to present receive higher weights, as recency is an important aspect of newsworthiness. While we have described our method for computing the `newsworthiness` value in some detail, we emphasize that for the rest of this article we only assume that the `newsworthiness` values are non-negative and that higher values indicate higher newsworthiness.

More crucially for the method described be-

low, we specify that the `value_type` fields (which describe how the messages' values are to be interpreted) contain members of a hierarchical taxonomy of data types represented as colon-separated hierarchies of labels. For example, the `value_type` field value `health:cost:hc2:mio_eur` would indicate that the number in the `value` field is the amount of money (`cost`), measured in millions of euros (`mio_eur`), spent by some nation (as defined by the `location` and `location_type` fields) on rehabilitative care (`hc2`) in some time period (as defined by the `timestamp` and `timestamp_type` fields) and that this is part of the larger health care topic (`health`). In our case, these labels are automatically established from the headers of the input data tables.

The goal of document structuring is to produce a three-level tree-structure with ordered children. The root node corresponds to the document as a whole and the mid-level structures correspond to paragraphs. The leaves are the messages selected for inclusion in the document. While the messages have not yet, at this stage, been associated with any linguistic structures, they can be conceptualized as being phrases or very short sentences. We are thus concurrently determining both the content and the structure the document.

We emphasize that our applications are employed in domains where they must be able to select some 10-30 messages from a pool of potential messages numbering in 100,000s. Given infinite computational resources, it would be preferential to construct all possible document plans and then score them in some fashion. This, however, is infeasible given the size of the search space. Previously, other authors have employed, for example, stochastic searches with significantly smaller search spaces (Mellish et al., 1998). Indeed, some kind of a beam search approach could be very useful in smartly searching a subset of the search space. However, we have thus far been unable to identify a document-level metric that adequately balances the 'total amount of newsworthiness' in a text with the length of the text, a requirement for beam search.

4 Research Objective

Based on the above considerations, our main goal is to identify a widely applicable method for content selection and document planning that matches the following requirements:

- REQ1: The method needs to be highly performant
- REQ2: The method should not be dependent on domain knowledge
- REQ3: The document should have a theme
- REQ4: The document should have multiple paragraphs but not be excessively long
- REQ5: The paragraphs should have distinct themes related to the document theme
- REQ6: The paragraph themes should be newsworthy in their own right
- REQ7: The paragraphs should not be excessively long or short
- REQ8: All messages should relate to the paragraph theme
- REQ9: All messages should be newsworthy
- REQ10: Within each paragraph, the messages should be presented in an order that produces a coherent narrative

Again, we emphasize that our goal is not to identify a method that is optimal for any specific scenario, but rather to determine a baseline method that is *adequate* for a broad spectrum of applications and sub-domains.

5 A Baseline Approach to Document Planning

Optimally, we would wish to produce some sort of a *globally optimal* document plan. However, as discussed above, this would entail significant computational costs and require a scoring function applicable to the document as a whole. As such, we propose a method for producing document plans in a greedy, linear, and iterative fashion. At every stage, decisions are made considering only a limited local context, thus avoiding the need for a method of determining the global quality of the document plan, thus fulfilling REQ1 ('The method needs to be highly performant').

The document's overall theme, in our use case, is selected by the user who initiates the generation task. In initiating the task, the users selects both a dataset and a focus location. The generation process then derives the *core messages* and *expanded messages* sets (the inputs to the Document Planner, see Section 3) so that both sets discuss the dataset

Field	Description	Example value
<code>where</code>	What location the fact relates to	Finland
<code>where_type</code>	What the type of the location is	country
<code>timestamp</code>	The time (or time range) the fact relates to	2020M05
<code>timestamp_type</code>	The type of the timestamp	month
<code>value</code>	A (usually) numeric value	0.01
<code>value_type</code>	Interpretation of <code>value</code>	<code>cphi:hicp2015:cp-hi02:rt01</code>
<code>newsworthiness</code>	An estimate of how newsworthy the message is	1

Table 1: An example of a message. The hypothetical message states that in the fifth month of 2020, in Finland, the consumer price index, using the year 2015 as the start of the index, of alcoholic beverages and tobacco changed by 0.01 points with respect to the value of the index during the previous month.

indicated by the user (i.e. messages from other datasets are not generated) and that the core set contains messages pertaining to the user’s indicated focus location, while messages pertaining to all other locations are in the expanded set. This fulfills REQ3 (‘The document should have a theme’). This step is also independent of the specific subdomain, thus fulfilling REQ2 (‘The method should not be dependent on domain knowledge’). This step thus fulfills all the relevant requirements. Next, we’ll describe how both the first and subsequent paragraphs can be planned in a way consistent with the requirements defined above.

5.1 Planning the First Paragraph

At the start of the document planning process, we select the most newsworthy message from the *core messages* set to act as the nucleus (n_1) of the first paragraph (p_1). This nucleus establishes the theme of the first paragraph as follows: We inspect the `value_type` field of this first nucleus n_1 , and retrieve a prefix `Prefix(n_1)`. The prefix is the least amount of colon-separated labels wherein the total amount of prefixes in the core set is greater than the minimal amount of paragraphs a document can have, in our case two. In our case, as a consequence of our label hierarchy, this is always the first three colon-separated units. For the message shown in Table 1, the prefix would thus be `cphi:hicp2015:cp-hi02`, meaning that the first paragraph’s theme would be the prices of alcoholic beverages and tobacco. This fulfills REQ5, ‘the paragraphs should have distinct themes related to the document theme’ for the first paragraph.

Next, the first paragraph is completed with satellites from the union of the *core messages* and the *expanded messages* sets. These satellites are initially filtered so that only messages that have the

same prefix as the nucleus n_i are considered in paragraph p_i to fulfill REQ8 (‘All messages should relate to the paragraph theme’). The satellites are then selected in a linear, greedy, and iterative manner to fulfill REQ1.

For selecting the k ’th satellite to a partially constructed paragraph already containing $k - 1$ satellites and one nucleus, we consider both the newsworthiness of the available messages (REQ9), as well as how well they would fit the already constructed segment (REQ8). Observing only the newsworthiness would produce a highly incoherent narrative, whereas focusing only on the narrative risks leaving out highly important information.

Following the reasoning in Section 2, we assume that two subsequent messages are more likely to form a good narrative if they are similar. As such, we need a method for weighing the message’s newsworthiness by the similarity of the message to the last message of the under-construction paragraph, thus balancing the requirements of REQ8 and REQ9. In terms of the message objects described in Table 1, it seems to us that the intuitive aspects of similarity are related to the degree of similarity within the ‘meta’ fields such as `timestamp`, `location` and `value_type`.

For the `timestamp` and `location` fields, we can state that two messages that have identical values in the fields are more similar than two messages that are otherwise the same but have distinct values for said fields. We call this the *contextual* similarity of the messages, and the fields the *contextual fields* (F_c), as these fields provide us access to the larger context in which the `value` and `value_type` fields can be interpreted. Contextual similarity captures the notion that it is likely better to follow a fact about French healthcare spending in 2020 with another piece of information about France in 2020,

rather than about Austria in 1990.

In more precise terms, we propose the following weighing scheme for contextual similarity: The similarity $sim_c(A, B)$ of two messages A and B is the product of weights $w_f > 1$ for each field f among the contextual fields F_c , where both A and B have the same value for the field:

$$sim_c(A, B) = \prod_{\{f \in F_c | A.f=B.f\}} w_f \quad (1)$$

This value strictly increases as more fields are shared between A and B . We explicitly define the similarity to be zero if there are no fields f where A and B share a value. If w_f is a uniform value for all fields f , this scheme is completely domain-agnostic. Setting different weights w_f for each field $f \in F_c$ allows for encoding some domain knowledge about which fields are the most important for the text, thus providing a method for producing more tailored texts at the cost of slightly violating REQ2. In our case study, we set $w_{timestamp} = 1.1$ and $w_{location} = 1.5$.

The above consideration of similarity still ignores valuable information available from the `value_type` field, which describes how the value in the `value` field is to be interpreted. Denoting `health:cost:hc2:mio_eur` (the cost of rehabilitative care in millions of euros) by T_1 , consider its similarity to $T_2 = \text{health:cost:hc2:eur_hab}$, the cost of rehabilitative care as euros per inhabitant, and $T_3 = \text{health:cost:hc41:mio_eur}$, the cost of health care related imaging services in millions of euros. Intuitively, T_1 and T_2 are thematically closer than T_1 and T_3 . We model this similarity between two facts A and B simply as

$$sim_t(A, B) = \frac{1}{s(A, B)} \quad (2)$$

where $s(A, B)$ is the length – in colon-separated units – of the unshared suffix between A and B 's `value_type` fields. That is, $s(T_1, T_2) = 1$ whereas $s(T_1, T_3) = 2$. We specify that $sim_t(\cdot, \cdot)$ is zero for all pairs without any shared prefix.

Our formulation of $sim_t(\cdot, \cdot)$ was influenced by the observation that in our context the messages' `value_type` values have a constant number of colon-separated segments. In cases where the lengths of the `value_type` values differ, an alternative formulation of

$$sim'_t(A, B) = \frac{2p(A, B)}{\ell(A) + \ell(B)} \quad (3)$$

where $\ell(\cdot)$ provides the length of the `value_type` value, and $p(\cdot, \cdot)$ is the length of shared *prefix* between A and B , both measured as colon-separated units, might be preferable if also more complex.

When considering whether the k 'th satellite s_i^k of paragraph p_i should be a specific candidate $c \in C$, where C is all so far unused messages, we can combine the similarity metrics with the newsworthiness of c into a general fitness value as follows:

$$\begin{aligned} fit(c, x) &= c.newsworthiness \\ &\times sim_c(c, x) \\ &\times sim_t(c, x) \\ &\times set_penalty(c) \end{aligned}$$

The $set_penalty(c)$ factor depends on whether the message originates from the *core messages* set, or the *extended messages* set. For messages originating from the core message set, the penalty is 1. For messages originating from the extended messages set, the penalty is $\frac{1}{dist+1}$, where $dist$ is the distance from the previous core message.

The final score describing how good of an addition c would be as the k th satellite of the i th paragraph s_i^k is then obtained by taking the average of fitnesses of c in relation to both the nucleus n_i and the previous satellite s_i^{k-1} by computing:

$$score(c, n_i, s_i^{k-1}) = \frac{fit(c, n_i) + fit(c, s_i^{k-1})}{2}$$

This maximizes the newsworthiness of the paragraph's contents (fulfilling REQ9, 'all messages should be newsworthy'), while also enforcing relatedness to the theme of the paragraph (fulfilling REQ8, 'all messages should relate to the paragraph theme') by measuring against the nucleus and with the inclusion of the $set_penalty$. By continuously measuring against the previously selected satellite, the procedure also allows for interludes to e.g. discuss highly newsworthy information related to but not strictly about the paragraph's main topic, or 'thematic drift'. It thus fulfills REQ10 ('Within each paragraph, the messages should be presented in an order that produces a coherent narrative') while also paying attention to the pyramid model of news (See Section 2).

Using $score$, the highest scoring candidate $c_{top} = \arg \max_{c \in C} score(c, n_i, s_i^{k-1})$ is then compared to both an absolute threshold t_{abs} and the newsworthiness of the nucleus n_i multiplied by relative threshold value t_{rel} . Provided that the

maximal paragraph length has not been reached, the top candidate message c_{top} is appended to the paragraph p_i as the k 'th satellite s_i^k in the document plan provided that either $score(c_{top}, n_i, s_i^{k-1}) \geq t_{abs}$ or $score(c_{top}, n_i, s_i^{k-1}) \geq t_{rel} \times n_i.newsworthiness$.

These thresholds ensure that the paragraph does not stray into minutiae, whether considered in absolute terms or in relation to the nucleus of the paragraph. In cases where the minimum paragraph length has not been reached, the thresholds are ignored and the top candidate is always appended. This accounts for REQ7 ('The paragraphs should not be excessively long or short').

The above considerations take into account several free parameters, namely the maximal and minimal paragraph lengths as well as the threshold values t_{rel} and t_{abs} . In our case study, we selected the minimal and maximal paragraph lengths as 2 and 5 messages empirically by trialing out various values and observing the resulting texts. These should, naturally, be based on the genre of text and the target audience. For the threshold values we selected 0.2 and 0.5, respectively, using the same method as with the paragraph lengths above. Both the thresholds and the minimal and maximal paragraph lengths should be viewed as (manually) tuneable hyperparameters.

5.2 Planning Subsequent Paragraphs

We then proceed to generate further paragraphs in a manner highly similar to that used when planning the first paragraph. The only distinction is that, when selecting the nucleus n_i for a subsequent paragraph p_i , we obtain the message from the *core messages* set with a highest newsworthiness value that has a prefix (theme) not yet discussed among the previously planned paragraphs $p_1 - p_{i-1}$:

$$n_i = \arg \max_{c \in C} c.newsworthiness \quad (4)$$

where

$$C = \left\{ c \in CoreMessages \mid \text{Prefix}(c) \notin \{ \text{Prefix}(n_k) \mid k \in [1..i-1] \} \right\} \quad (5)$$

This ensures that the different paragraphs are highly newsworthy, thus fulfilling REQ6, while also fulfilling REQ5 for having distinct themes for the different paragraphs.

As when constructing the subsequent paragraphs, the total length of the document also needs to

be considered. To fulfill REQ4 ('The document should have multiple paragraphs but not be excessively long'), we employ a variation of the method described in the previous section for ending individual paragraphs. A maximal length (in our case, 3 paragraphs) ensures that the document is not allowed to grow beyond reason, whereas a minimal length (for us, 2 paragraphs) ensures that the document is not unreasonably short. After the minimal length has been reached (but not yet the maximal length), a new paragraph is only started if the nucleus of the potential paragraph has a newsworthiness value that is at least 30 % of the newsworthiness value of the first nucleus of the document. This, as with the satellites, ensures that the the document does not stray into minutiae, balancing REQs 4 and 6. the maximal and minimal lengths, as well as the 30 % threshold, were determined by manual fine-tuning and should be viewed as tuneable hyperparameters.

6 Evaluation

The method described above was implemented in a larger NLG application producing news alerts for journalists from datasets provided by Eurostat. A variation of the same application was also developed with a simplified document planner. In this simplified planner, the planner always selects the maximally newsworthy available message as the message without any early stopping threshold. Nuclei are selected from the core messages set, while satellites can be from either set. Contrasting our proposed method with this simplified method enables us to evaluate the importance of narrative coherence in the generated texts. The larger application is multilingual, but the evaluation was conducted using English language texts.

Three experts were recruited from the Finnish News Agency STT, a national European news agency, to evaluate documents on the consumer price indices in five different European nations. For all nations, the judges were shown variants produced by both our proposed method and the simplified method. One of the selected countries is the country the news agency is based in, with the assumption that the judges would have high amounts of world knowledge they would be able to use in evaluating these texts. Another variant pair describes a country that is both relatively small and geographically remote (but still within EU), with the assumption that the journalists are unlikely to

Consumer Prices in Estonia

In June 2020, in Estonia, the monthly growth rate of the harmonized consumer price index for the category 'education' was 30.8 points. It was 30.7 percentage points more than the EU average. In July 2020, it was 0.4 percentage points less than the EU average. It was -0.4 points. In May 2020, the yearly growth rate of the harmonized consumer price index for the category 'education' was -20.5 points. It was 21.9 percentage points less than the EU average.

In August 2020, the monthly growth rate of the harmonized consumer price index for the category 'housing, water, electricity, gas and other fuels' was 2.5 points. It was 2.3 percentage points more than the EU average. In North Macedonia, it was 3 percentage points more than the EU average. It was 3.2 points. Estonia had the 3rd highest monthly growth rate of the harmonized consumer price index for the category 'housing, water, electricity, gas and other fuels' across the observed countries. In Sweden, the monthly growth rate of the harmonized consumer price index for the category 'housing, water, electricity, gas and other fuels' was 3.1 points.

Figure 1: Example output regarding Eurostat statistics on consumer prices. The text contains 12 messages, selected from among 207,210 messages available during generation.

have much world knowledge about this country's consumer prices. The three other countries were selected from among those bordering the first country, with the assumption that the journalists would have some, but not much, world knowledge relating to these countries. The final output texts were not inspected prior to selecting the countries.

All of the texts used in the evaluation were generated from a copy of the same underlying Eurostat dataset, entitled 'Harmonised index of consumer prices - monthly data [ei_cphi_m]² downloaded in September 2020. It contains country-level data regarding the harmonized consumer prices indices, and their change over time, for various EU nations starting from January 1996. We preprocess the data by adding monthly rankings (i.e. determine what country had the greatest, the second greatest, etc. value for a specific index category during any specific month) and comparisons to the EU average values.

As the evaluation was focused on document planning and content selection, the larger system was simplified in some respects, e.g., to not conduct

²Available for download and browsing from http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=ei_cphi_m

complex aggregation. This was done to minimize the effect of later stages of the generation process on the evaluation. As a result, the language in the evaluated documents was relatively stilted, as exemplified by Figure 1. The only manual alteration was the addition of headings to indicate the texts' intended themes.

The judges did not receive any direct compensation but their employer, the news agency, is a member of the EU-wide EMBEDDIA research project within which parts of this work was conducted. The evaluations were conducted online. The judges were first provided with some basic information on the type of documents they were to read (i.e. that the texts are intended to be news alerts for journalists, rather than publication ready news texts), the length of the task, etc. All instructions were in the judges' native language, in this case Finnish. The judges were not told which texts were produced by which variants nor how many variants were being tested. Following this, the judges were shown the documents one by one. For each document, the judges were asked to indicate their agreement with the following statements (translated from Finnish):

Q1: The text matches the heading

Q2: The text is coherent

Q3: The text lacks some pertinent information

Q4: The text contains unnecessary information

Q5: The text has a suitable length

For Q1–Q4, the judges indicated their agreement on a 7-point Likert scale ranging from 1 ('completely disagree') to 7 ('completely agree'). For Q5, the answers were provided on 5-point scale ranging from 1 ('clearly too short') to 3 ('length is suitable') to 5 ('clearly too long'). In addition, the judges were able to provide textual feedback for each individual text, as well as for the evaluation task as a whole. The judges' answers to Q1 – Q5, are aggregated in Table 2.

The results indicate that the proposed method statistically significantly increases the document's coherence (Q2, mean 4.33 vs. 1.60, median 5 vs 2), the matching of the document's content to the document's theme (Q1, mean 4.40 vs. 1.80, median 5 vs 2), and produces documents of more suitable length (Q5, mean 2.93 vs. 4.07, median 3 vs 4, with 3 being best). The proposed method also seems

Statement	Our method			Baseline			p_{MWU}
	Median	Mean	SD.	Median	Mean	SD.	
Q1 (1–7, ↑)	5	4.40	1.64	2	1.80	0.41	< 0.001*
Q2 (1–7, ↑)	5	4.33	1.76	2	1.60	0.51	< 0.001*
Q3 (1–7, ↓)	4	4.47	1.81	6	5.80	1.42	0.049
Q4 (1–7, ↓)	5	5.13	1.55	6	6.33	0.62	0.024
Q5 (1–5, 3 best)	3	2.93	0.59	4	4.07	0.70	< 0.001*

Table 2: Results obtained during the evaluation. Parentheses indicate answer ranges and whether the higher (↑), lower (↓) or middle values are to be interpreted as the best. The p_{MWU} column contains the (uncorrected) p-value of a two-sided Mann-Whitney U test. An asterisk indicates the p-value is statistically significant also after applying a Bonferroni correction to account for multiple tests.

to result in less unnecessary information being included in the document (Q4, mean 5.13 vs 6.33, median 5 vs 6), and in the text missing less necessary information (Q3, mean 4.47 vs 5.80, median 4 vs 6), but these effects are not statistically significant after correcting for multiple comparisons with the Bonferroni correction. We hypothesize this difference would become significant in a larger-scale evaluation.

The free-form textual feedback provided by the judges, as expected, indicates that the texts could be further improved. For example, in the case of the text shown in Figure 1, the judges called for a sentence explicitly noting that North Macedonia had the highest monthly growth rate. In addition, they noted it might be better to produce distinct, even shorter, texts as ‘news alerts’ while reserving the evaluated texts for use as a starting point when the journalist starts writing.

7 Conclusions

In this work, we have identified a need for, and proposed, a widely applicable baseline document planning method for generating journalistic texts from statistical datasets. Our method is based on observations on the similarities between the orbital theory of news structure (White, 1997) and Rhetorical Structure Theory (Mann and Thompson, 1988). While our proposed method is likely to fall short of the performance of subdomain-specific planning methods, results indicate that it achieves adequate performance while fulfilling a set of requirements identified based on the larger application domain of news generation.

Acknowledgements

This work is supported by the European Union’s Horizon 2020 research and innovation program under grant agreement No 825153, project EMBEDDIA (Cross-Lingual Embeddings for Less-Represented Languages in European News Media), and grant agreement No 770299, project NewsEye (A Digital Investigator for Historical Newspapers).

References

- Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512.
- Hadi Banaee, Mobyen Uddin Ahmed, and Amy Loutfi. 2013. Towards NLG for physiological data monitoring with body area networks. In *14th European Workshop on Natural Language Generation, Sofia, Bulgaria, August 8-9, 2013*, pages 193–197.
- David Caswell and Konstantin Dörr. 2018. Automated journalism 2.0: Event-driven narratives: From simple descriptions to real stories. *Journalism practice*, 12(4):477–496.
- Seniz Demir, Sandra Carberry, and Kathleen F. McCoy. 2010. A discourse-aware graph-based content-selection framework. In *Proceedings of the 6th International Natural Language Generation Conference*.
- Laurence Dierickx. 2019. Why news automation fails. In *Computation+ Journalism Symposium, Miami, FL*.
- Konstantin Nicholas Dörr. 2015. Mapping the field of algorithmic journalism. *Digital journalism*.
- Longxu Dou, Guanghui Qin, Jinpeng Wang, Jin-Ge Yao, and Chin-Yew Lin. 2018. Data2text studio: Automated text generation from structured data. In

- Proc. 2018 Conference on Empirical Methods in Natural Language Processing.*
- Ondřej Dušek, David M Howcroft, and Verena Rieser. 2019. Semantic noise matters for neural natural language generation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 421–426.
- Alexander Fanta. 2017. Putting Europe’s robots on the map: automated journalism in news agencies. *Reuters Institute Fellowship Paper*, pages 2017–09.
- Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Kraemer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. *arXiv preprint arXiv:1908.09022*.
- Johan Galtung and Mari Holmboe Ruge. 1965. The structure of foreign news: The presentation of the Congo, Cuba and Cyprus crises in four Norwegian newspapers. *Journal of peace research*, 2(1):64–90.
- Albert Gatt, Francois Portet, Ehud Reiter, Jim Hunter, Saad Mahamood, Wendy Moncur, and Somayajulu Sripada. 2009. From data to text in the neonatal intensive care unit: Using NLG technology for decision support and information management. *Ai Communications*, 22(3):153–186.
- Dimitra Gkatzia. 2016. Content selection in data-to-text systems: A survey. *arXiv preprint*. Available at <https://arxiv.org/abs/1610.08375>.
- Andreas Graefe. 2016. Guide to automated journalism.
- Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.
- Catalina Hallett, Richard Power, and Donia Scott. 2006. Summarisation and visualisation of e-health data repositories. In *UK E-Science All-Hands Meeting*.
- Jenna Kanerva, Samuel Rönqvist, Riina Kekki, Tapio Salakoski, and Filip Ginter. 2019. Template-free data-to-text generation of Finnish sports news. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 242–252, Turku, Finland. Linköping University Electronic Press.
- Ioannis Konstas and Mirella Lapata. 2013. Inducing document plans for concept-to-text generation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1503–1514.
- Leo Leppänen, Myriam Munezero, Mark Granroth-Wilding, and Hannu Toivonen. 2017a. Data-driven news generation for automated journalism. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 188–197.
- Leo Leppänen, Myriam Munezero, Stefanie Sirén-Heikel, Mark Granroth-Wilding, and Hannu Toivonen. 2017b. Finding and expressing news from structured data. In *Proceedings of the 21st International Academic Mindtrek Conference*, pages 174–183. ACM.
- Liunian Li and Xiaojun Wan. 2018. Point precisely: Towards ensuring the precision of data in generated texts using delayed copy mechanism. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1044–1055, Santa Fe, New Mexico, USA. ACL.
- Carl-Gustav Linden. 2017. Decades of Automation in the Newsroom: Why are there still so many jobs in journalism? *Digital Journalism*, 5(2):123–140.
- Veronika MacKová and Jakub Sido. 2020. The robotic reporter in the Czech News Agency: Automated journalism and augmentation in the newsroom. *Communication Today*, 11(1):36–53.
- William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.
- Chris Mellish, Alistair Knott, Jon Oberlander, and Mick O’Donnell. 1998. Experiments using stochastic search for text planning. In *Natural Language Generation*.
- Feng Nie, Jin-Ge Yao, Jinpeng Wang, Rong Pan, and Chin-Yew Lin. 2019. A simple recipe towards reducing hallucination in neural surface realisation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2673–2679.
- Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proc. 33rd AAAI Conference on Artificial Intelligence*.
- Horst Pöttker. 2003. News and its communicative quality: The inverted pyramid—when and why did it appear? *Journalism Studies*, 4(4):501–511.
- Ehud Reiter. 2007. An architecture for data-to-text systems. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 97–104. Association for Computational Linguistics.
- Ehud Reiter. 2018. Hallucination in neural NLG. <https://ehudreiter.com/2018/11/12/hallucination-in-neural-nlg/>. Accessed: 2020-03-02.
- Ehud Reiter. 2019. ML is used more if it does not limit control. <https://ehudreiter.com/2019/08/15/ml-limits-control/>. Accessed: 2020-07-25.
- Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Studies in Natural Language Processing. Cambridge University Press.

- Stefanie Sirén-Heikel, Leo Leppänen, Carl-Gustav Lindén, and Asta Bäck. 2019. Unboxing news automation: Exploring imagined affordances of automation in news journalism. *Nordic Journal of Media Studies*, 1(1):47–66.
- Somayajulu G Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. 2003. Generating English summaries of time series data using the Gricean maxims. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 187–196.
- Olli Sulopuisto. 2018. Uutisia kortti kerrallaan. *Suomen Lehdistö*. <https://suomenlehdisto.fi/uutisia-kortti-kerrallaan/>.
- Elizabeth A Thomson, Peter RR White, and Philip Kitley. 2008. “Objectivity” and “hard news” reporting across cultures: Comparing the news report in English, French, Japanese and Indonesian journalism. *Journalism studies*, 9(2):212–228.
- Peter White. 1997. Death, disruption and the moral order: the narrative impulse in mass-media ‘hard news’ reporting. *Genres and institutions: Social processes in the workplace and school*, 101:133.
- Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proc. 2017 Conference on Empirical Methods in Natural Language Processing*.
- Yleisradio. 2018. Avoin voitto. <https://github.com/Yleisradio/avoin-voitto>.
- Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. ACL.

Assessing the Quality of Human-Generated Summaries with Weakly Supervised Learning

Joakim Olsen and Arild Brandrud Næss

NTNU – Norwegian University of Science and Technology
Trondheim, Norway

joakiol@stud.ntnu.no, arild.naess@ntnu.no

Pierre Lison

Norwegian Computing Center
Oslo, Norway

plison@nr.no

Abstract

This paper explores how to automatically measure the quality of human-generated summaries, based on a Norwegian corpus of real estate condition reports and their corresponding summaries. The proposed approach proceeds in two steps. First, the real estate reports and their associated summaries are automatically labelled using a set of heuristic rules gathered from human experts and aggregated using weak supervision. The aggregated labels are then employed to learn a neural model that takes a document and its summary as inputs and outputs a score reflecting the predicted quality of the summary. The neural model maps the document and its summary to a shared “summary content space” and computes the cosine similarity between the two document embeddings to predict the final summary quality score. The best performance is achieved by a CNN-based model with an accuracy (measured against the aggregated labels obtained via weak supervision) of 89.5%, compared to 72.6% for the best unsupervised model. Manual inspection of examples indicate that the weak supervision labels do capture important indicators of summary quality, but the correlation of those labels with human judgements remains to be validated. Our models of summary quality predict that approximately 30% of the real estate reports in the corpus have a summary of poor quality.

1 Introduction

Many types of reports incorporate human-generated summaries that seek to highlight the most important pieces of information described in the

full document. This is notably the case for *real estate condition reports*, which are long, technical reports presenting the current condition (as it is known to the seller) of a property for sale, including the general state of each room, known damages and defects, and key technical aspects such as the heating, plumbing, electricity and roof. Despite the rich amount of information contained in these real estate reports, several surveys have shown that many buyers of real estate do not read the full documents but rather concentrate on the summaries (Sandberg, 2017). However, professionals regard the quality of these summaries as varying greatly, from good to very poor. Actors in the real estate market have suggested that this information deficit may play an important role in the reported 10% of Norwegian real estate transactions ending in conflict (Huseiernes Landsforbund, 2017).

In this work we explore ways of automatically measuring the quality of such summaries, using a corpus of 96 534 real estate condition reports and their corresponding summaries. Although there exists a substantial body of work on summary evaluation (Lloret et al., 2018), previous work has largely focused on automatically generated summaries, often by comparing those generated summaries to reference summaries written by humans. The automated evaluation of human-generated summaries, however, has received little attention so far.

This paper presents an approach to automatically evaluate the quality of human-generated summaries when no manually labelled data is available. Instead, we rely on a set of heuristic rules provided by domain experts to automatically annotate a dataset of summaries (each coupled to their full-length document) with quality indicators. Those annotations are subsequently aggregated into a single, unified annotation layer using weak supervision (Ratner et al., 2017, 2019), based on a generative model

that takes into account the varying coverage and accuracy of the heuristic rules.

Although one could in theory directly use the labels obtained through weak supervision as quality indicators for the summaries, such an approach has a number of limitations. Most importantly, heuristic rules are only triggered under certain conditions, and may therefore “abstain“ from providing a quality score on some summaries. For instance, we may have a rule stating that, if the full report describes a major defect or damage in the bathroom, then a summary that fails to mention this defect should be labelled as being of poor quality. This rule will only label summaries that meet this specific condition, and abstain from generating a prediction in all other cases. Some heuristic rules may also depend on the availability of external data sources that are not available at prediction time. For instance, one can exploit the fact that an insurance claim has been raised on the real estate as an indicator that the summary may have omitted to mention some important defects or damages. Needless to say, this heuristic can only be applied on historical data, and not on new summaries.

To address those shortcomings, we use the aggregated labels obtained via weak supervision as a stepping stone to train a neural model whose task is to assess the quality of a summary in respect to its full-length document. The neural model embeds both the document and its summary into a dedicated semantic space (referred to as the *summary content space*) and computes the final quality score using cosine similarity. As real estate condition reports are often long documents (10 pages or more), we conduct experiments with models based not only on embeddings of entire documents, but also on embeddings of sections, sentences and words.

The paper makes three contributions:

1. A framework to automatically (a) associate summaries with quality indicators based on expert-written rules, and (b) aggregate those indicators using weak supervision.
2. A neural model that predicts the summary quality by embedding both the document and its corresponding summary into a common summary content space, and then computing the similarity between the two vectors. The neural model is trained using the weakly supervised labels as described above.

3. An evaluation of this approach on a large corpus of Norwegian real estate condition reports and their associated summaries.

As detailed in Section 4, this weak supervision approach is able to outperform unsupervised methods based on Latent Semantic Analysis (Deerwester et al., 1990) or Doc2Vec embeddings (Le and Mikolov, 2014) – by a large margin. Although the approach is evaluated on a specific corpus of real estate reports, the proposed methodology can be applied to any type of summaries, provided human experts are able to specify heuristics to assess the summary quality in the target domain.

2 Related Work

2.1 Summary evaluation

Summary evaluation has so far been mostly studied in relation to the task of automatic text summarization, i.e., the automated generation of summaries conditioned on the full document (Rush et al., 2015; Cheng and Lapata, 2016; Gambhir and Gupta, 2017; Cao et al., 2018; Fernandes et al., 2019). However, few papers have investigated how to evaluate the quality of human-generated summaries such as the short summaries associated with real estate condition reports.

Lloret et al. (2018) provide an overview of evaluation metrics for text summarization, focusing on three quality criteria: *readability*, *non-redundancy* and *content coverage*. Although readability and non-redundancy are important criteria to evaluate automatic text summarization systems, they are less relevant for assessing human-generated summaries written by professionals. The criteria of content coverage is, however, relevant in both contexts, and will be the main focus of this paper.

Metrics for summary evaluation can be divided in three overarching groups (Cabrera-Diego and Torres-Moreno, 2018; Ermakova et al., 2019):

1. Manual evaluation based on human judgments, where participants fill questionnaires to rate the summary quality according to a number of criteria (Nenkova and Passonneau, 2004; Saggion et al., 2010).
2. Automatic evaluation from overlap-measures with reference summaries written by human experts (Lin, 2004; Conroy and Dang, 2008; Giannakopoulos, 2013; Zhang et al., 2020). One popular metric based on this idea is ROUGE (Lin, 2004), which is computed from

the proportion of n -grams that are observed in both the generated output and the reference summaries.

3. Automatic evaluation without reference summaries, typically using measures of divergence between the generated summary and the source document (Torres-Moreno et al., 2010; Louis and Nenkova, 2013; Cabrera-Diego and Torres-Moreno, 2018).

The evaluation method proposed in this paper fits into the last category, as we do not require the availability of reference summaries. However, contrary to divergence-based metrics, the summary quality is estimated here on the basis of heuristic rules provided by human experts.

2.2 Document similarity

The proposed approach is also related to models of semantic similarity, as the purpose of our summary evaluation is to assess the extent to which the criteria of content coverage is satisfied.

There is a vast body of existing work on how to measure the semantic similarity between documents. This topic is also the focus of various benchmarks, such as the Microsoft Research Paraphrase (MSRP) corpus (Dolan et al., 2004) and the Semantic Textual Similarity (STS) benchmark (Cer et al., 2017), both expressed as pairs of short documents. The ACL Anthology Network (Radev et al., 2009) is also used for measuring semantic similarity between articles in Liu et al. (2017). Gong et al. (2019) investigates how to measure similarity between documents of varying sizes.

Document similarity can be computed from topic models based on, e.g., Latent Dirichlet Allocation (Blei et al., 2003; Rus et al., 2013; Liu et al., 2017), or through document embeddings (Le and Mikolov, 2014; Lau and Baldwin, 2016; Liu et al., 2017; Cer et al., 2017; Gong et al., 2019; Vrbanc and Meštrović, 2020). Contextual word representations such as BERT, XLNet or GPT-3 (Devlin et al., 2018; Yang et al., 2019; Brown et al., 2020), can also be used to derive document embeddings and have been shown to improve performance on document similarity benchmarks (Reimers and Gurevych, 2019; Li et al., 2020), notably on the MSRP corpus and the STS benchmark.

Of particular relevance to this paper is the text matching approach of Zhong et al. (2020) in which the source document and potential summaries are

matched in a semantic space. Their approach is, however, optimised for the problem of extracting summaries, while our focus is on evaluating existing, human-generated summaries, using expert-written rules as quality indicators.

2.3 Weak supervision

The key idea behind weak supervision is to label data points using a combination of weak (noisy) supervision signals instead of relying on a single gold standard. Those supervision signals are typically expressed as *labeling functions*, which may take the form of heuristic rules, lookups in external knowledge bases, machine learning models, or even annotations from crowd-workers. The result of those labeling functions are then aggregated using a generative model that estimates the accuracy (and possible correlations) of each function. Once aggregated, the (probabilistic) labels can be employed to train any type of machine learning model using supervised learning. One key benefit of weak supervision frameworks lies in their ability to inject *expert knowledge* to learn data-driven models in situations when data is scarce or non-existent (Hu et al., 2016; Wang and Poon, 2018).

Weak supervision makes it possible to leverage external knowledge sources to automatically label data points instead of relying exclusively on hand-annotated data. An early application of this idea is distant supervision (Mintz et al., 2009; Ritter et al., 2013), where knowledge bases are used to automatically label documents with specific categories. One popular approach for weak supervision is the Snorkel framework, which was first introduced by Ratner et al. (2016), and later expanded by Ratner et al. (2017) and Ratner et al. (2019).

Weak supervision frameworks have been applied to a number of NLP tasks, from named entity recognition to relation extraction and dialogue state tracking (Bach et al., 2019; Bringer et al., 2019; Hancock et al., 2019; Lison et al., 2020; Safranchik et al., 2020). There is, however, little work with weak supervision related to document similarity or summary quality evaluation.

3 Approach

The approach adopted in this paper is divided in two steps. We first define and apply a set of *labeling functions* to the dataset, allowing us to derive binary (good/bad) quality indicators on the summaries in relation to their full-length reports. Those

quality indicators are then aggregated into a single, probabilistic measure of summary quality using weak supervision. The dataset and labeling functions are described in Sections 3.1 and 3.2.

Then, using those aggregated labels as targets, we learn a neural model that maps the reports and summaries to a common *summary content space*. The resulting embeddings should reflect only key semantic information that is relevant for measuring summary quality, so that it can be measured by the cosine similarity in this space. The neural architecture and associated document embedding methods are defined in Sections 3.3 and 3.4.

Assessing the summary quality using a neural model instead of relying directly on the quality indicators derived from the labeling functions has two major advantages. First, the neural model can generalise to all possible report/summary pairs, while aggregated labels may be absent for some summaries, as the rules are only triggered when specific conditions are met. Second, some labeling functions depend on external resources that may be unavailable at prediction time. For instance, one labeling function relies on whether the buyer has filed an insurance claim, which is a piece of information that is only available for historical data, and requires us to “peek into the future”.

3.1 Dataset

The corpus contains 96 534 real estate condition reports, each containing the following parts:

- i) Textual descriptions of various parts of the real estate (e.g., rooms) along with a textual assessment of their physical condition.
- ii) Condition degrees (“tilstandsgrad” or TG) for parts of the real estate, in the range 0–3, where 0 indicates perfect condition (for new buildings) and 3 a seriously deteriorated condition, due to a major damage or defect.
- iii) Metadata for the real estate and the condition report – e.g., size, building year, the author of the report, date of assessment, etc.
- iv) The summary.

We consider (i) as constituting the full-length report, denoted r , while the summary text (iv) will be denoted s . The metadata (ii)–(iii) is used only by the weak supervision model. The average report length is 1287 words (standard deviation: ± 627 words), while the average summary length is 183 words (standard deviation: ± 138 words).

3.2 Labeling Functions

A collection of 22 labeling functions was specified in cooperation with domain experts. Each function has two possible output values, depending on whether it implies a bad summary, denoted by (–) or a good summary, denoted by (+). If the rule condition is not met, the rule abstains from suggesting an output (Ratner et al., 2017). The full list of labeling functions is the following:

1. Summary shorter than 50 words. (–)
2. Summary longer than 400 words. (–)
3. TG3 for the bathroom, but no mention of the bathroom in summary. (–)
4. TG3 for the kitchen, but no mention of the kitchen in summary. (–)
5. TG3 for the roof, but no mention of the roof in summary. (–)
6. TG2 or TG3 for the bathroom, with mention of the bathroom in summary. (+)
7. TG2 or TG3 for the kitchen, with mention of the kitchen in summary. (+)
8. TG2 or TG3 for the roof, with mention of the roof in summary. (+)
9. Correction of TG in the bathroom, but no mention of the bathroom in summary. (–)
10. Correction of TG in the kitchen, but no mention of the kitchen in summary. (–)
11. Correction of TG on the roof, but no mention of the roof in summary. (–)
12. Summary with long words readability score (LIKS) above 55. (–)
13. Summary with unique words readability score (OVR) above 96. (–)
14. An insurance claim has been raised on the real estate after the transaction. (–)
15. Written by an agent with insurance claims on more than 7.5% of her reports. (–)
16. Written by an agent with LIKS-score higher than 55 on more than 40% of her reports. (–)
17. Written by an agent with OVR-score higher than 96 on more than 40% of her reports. (–)
18. Written by an agent with fewer than 10 reports that year. (–)
19. Fewer than 20% of the words in the summary are found in the report. (–)
20. Fewer than 3% of the words in the report are found in the summary. (–)
21. More than 70% of the words in the summary are also found in the report. (+)
22. More than 20% of the words in the report are also found in the summary. (+)

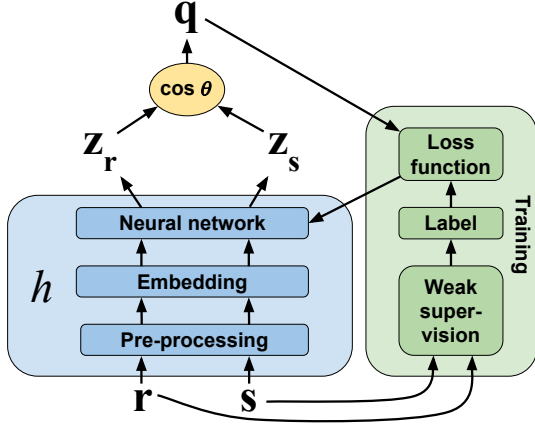


Figure 1: General model architecture $q(\mathbf{r}, \mathbf{s})$.

For a given summary, let y be the unknown true label, with possible values -1 (bad) and 1 (good), and let λ be the outputs of the labeling functions. By applying these to the real estate condition reports, a generative label model $P_\mu(y | \lambda)$ can be estimated in a fully unsupervised fashion, as described by Ratner et al. (2019). We then obtain labels $y^+ = P_\mu(y = 1 | \lambda) \in [0, 1]$, indicating the probability that a given summary is good.

3.3 Summary Quality Model

Let \mathcal{R} denote the set of all possible reports and summaries, and let \mathcal{Z} be the summary content space. We define the summary quality model as a function $q(\mathbf{r}, \mathbf{s})$ comparing two document embeddings:

$$q(\mathbf{r}, \mathbf{s}) = \cos \text{sim}(h(\mathbf{r}), h(\mathbf{s})) = \cos \text{sim}(\mathbf{z}_r, \mathbf{z}_s),$$

where $h : \mathcal{R} \rightarrow \mathcal{Z}$ is a learned mapping from texts (full reports or summaries) to vectors. The general architecture is illustrated in Figure 1.

The training objective for h should be such that a good (bad) summary should yield a high (low) cosine similarity. We also want our models to return quality scores distributed over the entire cosine domain $[-1, 1]$, and we find that the standard cross-entropy loss tends to push the values towards the edges. Instead, we use a variation of the cosine embedding loss function, given by

$$l(q(\mathbf{r}, \mathbf{s}), y) = \begin{cases} \max(0, \tau_{\text{good}} - \cos \text{sim}(\mathbf{z}_r, \mathbf{z}_s)), & y = 1 \\ \max(0, \cos \text{sim}(\mathbf{z}_r, \mathbf{z}_s) - \tau_{\text{bad}}), & y = -1, \end{cases}$$

where τ_{good} and τ_{bad} are thresholds on the quality scores of good/bad summaries. A loss of zero is obtained if good summaries have a quality score

higher than τ_{good} or if bad summaries have a quality score lower than τ_{bad} . The model will thereby not perform better by pushing the quality of summaries above τ_{good} or below τ_{bad} , which encourages the model to return scores on a larger part of the cosine domain $[-1, 1]$. We find experimentally that $\tau_{\text{good}} = 0.2$ and $\tau_{\text{bad}} = -0.2$ result in models with an appropriate distribution of values.

The weak supervision labels y^+ are expected to be noisy. We follow Ratner et al. (2019) in using a noise-aware version of our loss function $l(q(\mathbf{r}, \mathbf{s}), y)$ for training, which we define by

$$l^*(q(\mathbf{r}, \mathbf{s}), y^+) = E_{y \sim P_\mu(y | \lambda)} [l(q(\mathbf{r}, \mathbf{s}), y)] \quad (1) \\ = y^+ \cdot l(q(\mathbf{r}, \mathbf{s}), 1) + (1 - y^+) \cdot l(q(\mathbf{r}, \mathbf{s}), -1).$$

Having defined the general model architecture and its training procedure, we now detail various solutions to express the mapping h .

3.4 Document embeddings

3.4.1 LSA and Doc2vec

We start with unsupervised baseline models, and experiment with both Latent Semantic Analysis (Deerwester et al., 1990) and Doc2vec (Le and Mikolov, 2014), for their ability to easily embed arbitrarily long documents. We train LSA and Doc2vec on the training set (ignoring the quality labels, as those techniques are self-supervised). These models can be described in Figure 1 by removing the training and neural network components, and by using LSA or Doc2vec for the embeddings. We use a dimensionality of 500 for LSA and 100 for Doc2vec.

3.4.2 FFN-based models

Our first supervised model for h is a feed-forward network. We first embed the reports and summaries with LSA or Doc2vec (both of dimension 500) as described above, and add a feed-forward transformation of those vectors which is optimised on the basis of the embedding loss function. The network weights are shared for both the full report \mathbf{r} and the summary \mathbf{s} . The architecture becomes as illustrated in Figure 1 by inserting LSA or Doc2vec into the embedding component and a feed-forward network into the neural-network component. We refer to the resulting models as LSA+FFN and Doc2vec+FFN.

We employ the ReLU activation function in all layers except the last, which is linear (i.e., has no activation function). By using only a single feed-forward layer, this model architecture becomes

equivalent to a linear transformation of the LSA or Doc2vec embeddings. We refer to the resulting models as LSA+LinTrans and Doc2vec+LinTrans.

The hidden layers have 1000 units, and the final layer 100 units. LSA+FFN and Doc2vec+FFN respectively use two and three hidden layers.

3.4.3 LSTM-based models

The second model for the function h mapping reports and summaries to the summary content space is an LSTM network. LSTMs are commonly used over word embeddings, but this approach is hard to scale due to the length of real estate condition reports. Instead, we split the reports into sections, and summaries into sentences and use LSA or Doc2vec to embed each, giving a sequence of vectors for each report and summary, and train the LSTM on these. A final, fully connected linear layer is placed on the LSTM output. In Figure 1 the pre-processing component now includes the splitting of sections/sentences, the embedding component is LSA or Doc2vec, and the neural-network component is the LSTM. We refer to the resulting models as LSA+LSTM and Doc2vec+LSTM. We use a single, unidirectional LSTM layer with a cell dimensionality of 100, along with 100 units in the final dense layer.

3.4.4 Convolutional models

The final model for h is a convolutional neural network with word embeddings as inputs. Those word embeddings are estimated either by Word2vec (dimension: 100) or a neural embedding layer (dimension: 500), both trained on the training set of the corpus. We use 1D convolutions with window size $\in \{2, 3, 5, 7, 10\}$ and a number of filters equivalent to the word embedding dimension. We then apply a maximum pooling to obtain a single output vector, fed to a final, fully-connected linear layer.

One benefit of convolutional neural networks is their scalability when processing long documents. The convolutional model detects local text patterns that are especially predictive for the summary quality, thereby providing a good mapping to the summary content space. In Figure 1 the pre-processing component now includes tokenisation, the embedding component is the embedding layer or Word2vec, and the neural-network is the CNN. We refer to the resulting models as EmbLayer+CNN and Word2vec+CNN.

No.	Cov.	Overlap	Conflict	Acc.
1 (−)	10.4 %	96.2 %	22.1 %	100 %
2 (−)	7.9 %	91.1 %	82.3 %	10.9 %
3 (−)	5.1 %	90.2 %	27.5 %	71.5 %
4 (−)	2.4 %	95.8 %	50.0 %	58.5 %
5 (−)	2.6 %	92.3 %	30.8 %	78.0 %
6 (+)	36.9 %	76.4 %	46.1 %	74.9 %
7 (+)	11.6 %	93.1 %	47.4 %	97.3 %
8 (+)	25.1 %	83.7 %	46.2 %	82.0 %
9 (−)	7.6 %	84.2 %	22.4 %	73.5 %
10 (−)	5.1 %	90.2 %	45.1 %	60.8 %
11 (−)	8.1 %	82.7 %	34.6 %	72.9 %
12 (−)	11.8 %	92.4 %	42.4 %	73.4 %
13 (−)	10.7 %	93.5 %	26.2 %	100 %
14 (−)	1.8 %	83.3 %	55.6 %	47.9 %
15 (−)	1.6 %	93.8 %	43.8 %	71.5 %
16 (−)	10.8 %	88.9 %	48.1 %	57.9 %
17 (−)	10.0 %	91.0 %	37.0 %	100 %
18 (−)	5.4 %	85.2 %	48.1 %	58.9 %
19 (−)	3.4 %	76.5 %	14.7 %	83.4 %
20 (+)	6.3 %	85.7 %	49.2 %	63.3 %
21 (−)	7.1 %	94.4 %	11.3 %	100 %
22 (+)	6.2 %	91.9 %	48.4 %	100 %

Table 1: Analysis of the 22 labeling functions when applied to the real estate condition report corpus.

4 Evaluation

4.1 Weak Supervision Labels

Table 1 shows for each labeling function its coverage (as a percentage of the full corpus), the proportion of overlaps with at least one other labeling function, the proportion of conflicts with at least one other labeling function, and its accuracy estimated through the aggregated label model.

The weak supervision model abstains from labeling 15.9% of the summaries, giving us a labeled dataset of $M_{\text{lab}} = 81\,195$ samples. Figure 2 shows a histogram of the resulting probabilistic labels, $y_m^+ = P_{\mu}(y_m = 1 \mid \lambda_m)$ for $m = 1, \dots, M_{\text{lab}}$, where each y_m^+ is the probability of summary m being of high quality. We observe many summaries for which $y_m^+ \approx 0$ or $y_m^+ > 0.7$. The labels seem otherwise quite evenly distributed on the probability range $[0, 1]$, and their average is 0.493, which indicates that the dataset is well balanced and does not require oversampling. We split the labeled dataset of 81 195 samples in the ratio 8:1:1, yielding a training set of 64 955 samples and validation and test sets of 8 120 samples each.

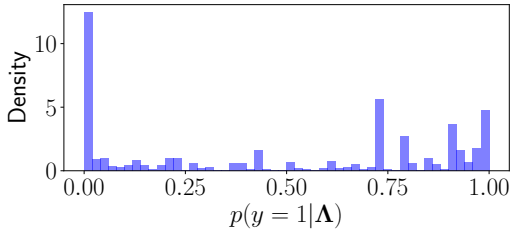


Figure 2: Histogram showing the distribution of labels from the weak supervision model.

4.2 Model Performance

We evaluate our models against the weak supervision labels. The model performances on the test set are given in Table 2, measured by the standard classification scores accuracy and F_1 , and for the supervised models also by the loss function given in (1). We train the models using the Adam optimizer with a learning rate of 1×10^{-4} , reduced by a factor of 0.1 after one third of the epochs, and again after two thirds. We also employ a dropout of 0.2 in the hidden layers.

For the computation of accuracy and F_1 , the probabilistic labels y^+ and the quality measures $q(\mathbf{r}, \mathbf{s})$ are converted to binary labels; the threshold for y^+ is 0.5, while for $q(\mathbf{r}, \mathbf{s})$ the threshold is tuned on the validation set. We see that the supervised models outperform the unsupervised ones and that the model Word2vec+CNN achieves the best performance both in terms of accuracy and F_1 .

It should be noted that the aggregated labels obtained with weak supervision only constitute a proxy for the ground truth. Although we expect them to provide good indications of the overall quality of the summaries in this domain, we cannot be certain of how well they correlate with human judgment, so our conclusions regarding the ability of various models to measure summary quality must remain somewhat tentative.

Figure 3 illustrates the performance of four models by showing the distributions of quality measures for samples where the weak supervision label model is confident about the label. Summaries with $y^+ \geq 0.9$ are shown in green and those with $y^+ \leq 0.1$ are shown in red. We observe that all of these models are, to some degree, able to distinguish good summaries from bad ones. The unsupervised LSA baseline does, however, have much more overlap than the other models, which reflects the poorer performance in Table 2. The distributions for the model LSA+LSTM is unexpected,

Model	Loss	Acc.	F_1
LSA	-	0.726	0.755
Doc2vec	-	0.684	0.686
LSA+LinTrans	0.095	0.863	0.876
Doc2vec+LinTrans	0.101	0.850	0.863
LSA+FFN	0.080	0.882	0.893
Doc2vec+FFN	0.079	0.885	0.897
LSA+LSTM	0.079	0.882	0.895
Doc2vec+LSTM	0.080	0.880	0.891
EmbLayer+CNN	0.088	0.888	0.898
Word2vec+CNN	0.085	0.895	0.905

Table 2: Model performances on the test set.

in that it pushes the quality measures just below $\tau_{\text{bad}} = -0.2$ or just above $\tau_{\text{good}} = 0.2$, instead of distributing them on the complete quality range $[-1, 1]$. This behavior effectively makes it a classifier rather than a model of quality measure. We observe the same behavior for the Doc2vec+LSTM model and FFN-based models. The LinTrans and CNN-based models, on the other hand, yield a good separation of good and bad summaries, while distributing them on a large portion of the quality range, which is the behavior we seek.

Figure 4 illustrates the distribution of quality measures assigned to all of the $M = 96\,534$ samples in the corpus by the Word2vec+CNN model. By comparing this histogram to the one in Figure 2, we see that this model provides a more continuous quality measure than the labels aggregated from the labeling functions using weak supervision.

4.3 Summary Quality

When applied to the entire corpus of real estate condition reports and summaries, including the ones that the weak supervision model abstained from labeling, the Word2vec+CNN model finds that 35% of the summaries have a quality score $q(\mathbf{r}, \mathbf{s})$ below $\tau_{\text{bad}} = -0.2$, our chosen threshold for being of poor quality, while 33% are judged to be of high quality (i.e., $q(\mathbf{r}, \mathbf{s}) > \tau_{\text{good}} = 0.2$), while the remaining 31% are considered mediocre. The LSA+LinTrans model find 28% of the summaries to be of poor quality, and an average of the CNN and LinTrans models gives a proportion of poor summaries around 30%. If almost a third of the summaries of real estate condition reports are in fact of poor quality, this would bode ill for the real estate buyers that do not read the full reports.

Three example summaries are included in Ap-

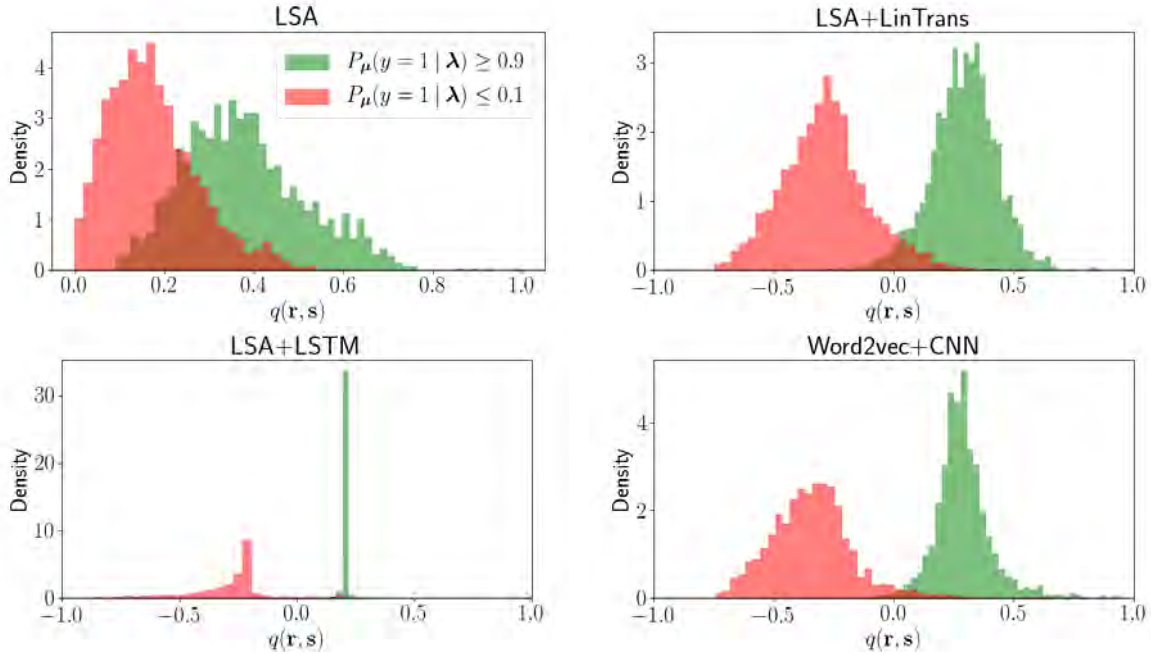


Figure 3: Normalized histograms showing the distribution of quality measures $q(\mathbf{r}, \mathbf{s})$ for summaries from the test set that the label model considers as good (shown as green) and bad (shown as red).

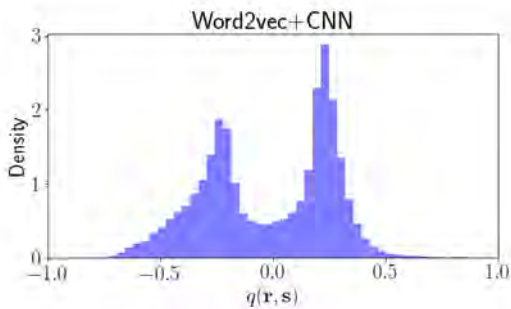


Figure 4: Normalized histogram of $q(\mathbf{r}, \mathbf{s})$ for the entire corpus of summaries.

pendix A. Their predicted quality measures using the weak supervision model, the LinTrans models and the CNN models are given in Table 3. We see that all models agree that the first summary is of good quality, and that the second is relatively bad. Since the first summary is quite thorough while the second is excessively short and quite uninformative, this is in line with our expectations. The third summary, however, is considered poor by the label model but quite good by the neural models. As this is also a quite thorough summary which captures the essence of its corresponding report, it would seem that the supervised models outperform in this case the labels they were trained on. We observed several such examples in the corpus, but without data from human judgments, we cannot ascertain

Model	Ex. 1	Ex. 2	Ex. 3
$P_{\mu}(y = 1 \lambda)$	0.92	0	0
LSA+LinTrans	0.24	-0.68	0.32
Doc2vec+LinTrans	0.46	-0.54	0.28
EmbLayer+CNN	0.67	-0.62	0.41
Word2vec+CNN	0.23	-0.68	0.61

Table 3: Quality scores for the three example summaries given in the appendix.

to what extent the neural models are truly more reliable than the weak supervision labels.

5 Conclusion

This paper describes a novel approach to automatically assess the quality (focusing primarily on the criteria of content coverage) of human-generated summaries, using a corpus of real estate condition reports as a concrete example. The approach relies on the creation of document embeddings that are appropriate for measuring summary quality. This gives us a particular kind of semantic space (the summary content space) where summary quality can be measured by the cosine similarity between the report and its summary.

Since we have no access to “ground truth” values for the summary quality, we obtain indirect quality indicators based on a set of 22 heuristic rules gathered from human experts. Those quality indi-

cators are then aggregated into a single probability (of a summary being of high quality) using weak supervision. The aggregated probabilities are subsequently employed as targets for training neural models optimised for the task of predicting summary quality. Evaluation results show that the best neural model, based on a convolutional architecture, achieves an overall accuracy of 89.5% when measuring the model output against the aggregated labels, while the best unsupervised model (LSA) only achieves an accuracy of 72.6%.

An important limitation of the proposed method is the reliance on indirect indicators of summary quality (as expressed by the heuristic rules) instead of human judgments. A key research question for future work is thus to examine the correlations between the quality measures derived from the labeling functions and human judgments. While the heuristic rules do not capture all aspects that may influence the overall quality of a summary, our hypothesis (yet to be validated) is that they nevertheless correlate well with human judgments. An additional benefit of these heuristic rules is their explanatory power, making it possible to provide concrete, human-readable suggestions on *how* to improve a given summary.

Although not considered in this paper, the use of document embeddings relying on contextual word representations is another interesting research question that we wish to investigate in future work.

References

- Stephen H. Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alex Ratner, Braden Hancock, Houman Alborzi, Rahul Kuchhal, Chris Ré, and Rob Malkin. 2019. Snorkel DryBell: A case study in deploying weak supervision at industrial scale. In *Proceedings of the 2019 International Conference on Management of Data*, pages 362–375. Association for Computing Machinery.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.
- Eran Bringer, Abraham Israeli, Yoav Shoham, Alex Ratner, and Christopher Ré. 2019. Osprey: Weak supervision of imbalanced extraction problems without code. In *Proceedings of the 3rd International Workshop on Data Management for End-to-End Machine Learning*, pages 1–11. Association for Computing Machinery.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Luis Adrián Cabrera-Diego and Juan-Manuel Torres-Moreno. 2018. Summtriver: A new trivergent model to evaluate summaries automatically without human references. *Data & Knowledge Engineering*, 113:184–197.
- Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 152–161, Melbourne, Australia. Association for Computational Linguistics.
- Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14. Association for Computational Linguistics.
- Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.
- John M. Conroy and Hoa Trang Dang. 2008. Mind the gap: Dangers of divorcing evaluations of summary content from linguistic quality. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, pages 145–152, Manchester, UK. COLING 2008 Organizing Committee.
- Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on*

- Computational Linguistics*, pages 350–356. Association for Computational Linguistics.
- Liana Ermakova, Jean Valère Cossu, and Josiane Mothe. 2019. A survey on evaluation of summarization methods. *Information processing & management*, 56(5):1794–1814.
- Patrick Fernandes, Miltiadis Allamanis, and Marc Brockschmidt. 2019. Structured neural summarization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66.
- George Giannakopoulos. 2013. Multi-document multilingual summarization and evaluation tracks in ACL 2013 MultiLing workshop. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, pages 20–28, Sofia, Bulgaria. Association for Computational Linguistics.
- Hongyu Gong, Tarek Sakakini, Suma Bhat, and Jinjun Xiong. 2019. Document similarity for texts of varying lengths via hidden topics. *arXiv preprint arXiv:1903.10675*.
- Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3684, Florence, Italy. Association for Computational Linguistics.
- Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany. Association for Computational Linguistics.
- Huseiernes Landsforbund. 2017. Konfliktnivået ved bolighandel må ned. [Online; accessed 3-February-2021].
- Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*.
- Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. Proceedings of Machine Learning Research.
- Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.
- Pierre Lison, Jeremy Barnes, Aliaksandr Hubin, and Samia Touileb. 2020. Named entity recognition without labelled data: A weak supervision approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1518–1533, Online. Association for Computational Linguistics.
- Ming Liu, Bo Lang, Zepeng Gu, and Ahmed Zeeshan. 2017. Measuring similarity of academic articles with semantic profile and joint word embedding. *Tsinghua Science and Technology*, 22(6):619–632.
- Elena Lloret, Laura Plaza, and Ahmet Aker. 2018. The challenging task of summary evaluation: An overview. *Language Resources and Evaluation*, 52(1):101–148.
- Annie Louis and Ani Nenkova. 2013. Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39(2):267–300.
- Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.
- Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.
- Dragomir R. Radev, Pradeep Muthukrishnan, and Vahed Qazvinian. 2009. The ACL Anthology network. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries (NLP4DL)*, pages 54–61, Suntec City, Singapore. Association for Computational Linguistics.
- Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3):269–282.
- Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. 2019. Training complex models with multi-task weak supervision. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:4763–4771.

- Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3574–3582. Curran Associates Inc.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *arXiv preprint arXiv:1908.10084*.
- Alan Ritter, Luke Zettlemoyer, Mausam, and Oren Etzioni. 2013. Modeling missing data in distant supervision for information extraction. *Transactions of the Association for Computational Linguistics*, 1:367–378.
- Vasile Rus, Nobal Niraula, and Rajendra Banjade. 2013. Similarity measures based on latent dirichlet allocation. In *Computational Linguistics and Intelligent Text Processing*, pages 459–470. Springer.
- Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.
- Esteban Safranchik, Shiyong Luo, and Stephen Bach. 2020. Weakly supervised sequence tagging from noisy rules. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5570–5578.
- Horacio Saggion, Juan-Manuel Torres-Moreno, Iria da Cunha, Eric SanJuan, and Patricia Velázquez-Morales. 2010. Multilingual summarization evaluation without human models. In *COLING 2010: Posters*, pages 1059–1067, Beijing, China. COLING 2010 Organizing Committee.
- Tor Sandberg. 2017. Kjøper dyre boliger i blinde. *Dagsavisen*. [Online; accessed 3-February-2021].
- Juan-Manuel Torres-Moreno, Horacio Saggion, Iria da Cunha, Eric SanJuan, and Patricia Velázquez-Morales. 2010. Summary evaluation with and without references. *Polibits*, (42):13–20.
- Tedo Vrbanec and Ana Meštrović. 2020. Corpus-based paraphrase detection experiments and review. *Information*, 11(5):241.
- Hai Wang and Hoifung Poon. 2018. Deep probabilistic logic: A unifying framework for indirect supervision. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1891–1902, Brussels, Belgium. Association for Computational Linguistics.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763. Curran Associates, Inc.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.
- Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

Appendix A. Example Summaries

1. Enebolig fra 1978 som er holdt vedlike og har god standard, tatt alder i betraktning. Den er noe påkostet over tid ellers er det originalt. Det er valmtak med bordtak. Renner og nedløp. Bindingsverkvegger som er isolert med stående panel og murforblending. Vinduer med karm og ramme i tre med isolerglass. Massiv utgangsdør i teak. Det er leca grunnmur og støpt dekke. Dreneringen er fra byggetiden. Innvendig er det panel og plater i himling, gulv har fliser, belegg, laminat, teppe og parkett. Baderom med fliser på gulv og vegger med sanitær utstyr som er fra byggetiden. Det er eget wc rom og dusjkabinett i fyr-rom og wc med servant i vaskerom. Eik kjøkkeninnredning med profiler på overskap og underskap fra byggetiden. Sentralfyr for olje og strøm som er ca 10 år. Oljetank under terrasse. Elektrisk anlegg med skrusikringer. Garasje fra 1986 den er oppført med støpt dekke, leca ringmur, stående kledning. Valmtak med betongstein, renner og nedløp i plastbelagt stål. Det er 2 stk leddporter. Det er registrert vanlig elde og bruksslitasje på eiendommen.
2. Boligen ligger i et etablert boligområde, med kort vei til skole, barnehage og forretning. Det er gjort bemerkninger som bør utbedres, som våtrom og oppgraderinger pga. normal bruksslitasje. Forøvrig les rapport.
3. Bolig bygget i år 2005 med gjeldende forskrifter fra byggeår. (Plan og bygningsloven fra 1985, revidert i 1997. Teknisk forskrift -97.) Boligen og garasje fremstår som normalt vedlikeholdt. Malte flater på alle vegger og himlinger i oppholdsrom. Keramiske fliser på gulv og vegger i bad. Keramiske fliser på gulv i vaskerom. Vedovn i stue med inndekning fra år 2010. Gruset

område rundt boligen. Stor terrasse på oppside med støpte fundamenter. Garasje med plass til to biler. Keramiske fliser på vegger og gulv i bad. TG2 grunnet alder. Keramiske fliser på gulv i vaskerom. Vegger platet med malt tapet. TG2 grunnet alder. Adkomstdør trenger justering. TG2 Platon grunnmursplate. Manglende topp-list. Dette kan samle fukt mot grunnmur. Løv og barnåler bak platonplate ble registrert ved befaring. Rensing og festing av plate anbefales. TG3 Ett nedløp i front av bolig ikke tilkoblet drenerør. TG2.

Knowledge Distillation for Swedish NER models: A Search for Performance and Efficiency

Lovisa Hagström

Chalmers University of Technology
Sweden

lovhag@chalmers.se

Richard Johansson

University of Gothenburg
Sweden

richard.johansson@cse.gu.se

Abstract

The current recipe for better model performance within NLP is to increase model size and training data. While it gives us models with increasingly impressive results, it also makes it more difficult to train and deploy state-of-the-art models for NLP due to increasing computational costs. Model compression is a field of research that aims to alleviate this problem. The field encompasses different methods that aim to preserve the performance of a model while decreasing the size of it. One such method is knowledge distillation. In this article, we investigate the effect of knowledge distillation for named entity recognition models in Swedish. We show that while some sequence tagging models benefit from knowledge distillation, not all models do. This prompts us to ask questions about in which situations and for which models knowledge distillation is beneficial. We also reason about the effect of knowledge distillation on computational costs.

1 Introduction

Currently, most research that pushes the boundary for state-of-the-art performance within natural language processing involves the increase of number of model parameters as well as the computations needed for training (Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020). The trend seems to be that the larger the model, the better the performance. As noted by Strubell et al. (2019) these state-of-the-art models require significant computational resources during training as well as deployment. While it certainly is a good thing that state-of-the-art performance within NLP is continuously improving, there is work to be

done on model efficiency. More efficient models are needed both for the sake of the environment and for the sake of equal research opportunities. Here we define an “efficient model” based on both performance and computational cost, such that a model is more efficient if it has better performance or lower computational cost, and vice versa.

Knowledge distillation (Hinton et al., 2015) is one way to improve model efficiency during deployment. There are several works on successful application of knowledge distillation both for pre-training tasks and for specific downstream tasks. Adhikari et al. (2020) show that knowledge distillation can be used to improve deployment efficiency of models for the downstream task of document classification in English.

In this article we investigate the effect of knowledge distillation on models for named entity recognition (NER) in Swedish.¹ The intention is to shed some light on how well knowledge distillation performs for different sequence tagging models and in the Swedish language. Our main goal is to contribute to better model efficiency within NLP. Naturally, this entails that we also focus on measuring the efficiency of each model investigated. Hopefully, this work will facilitate the development of more efficient models for both the English and the Swedish language.

2 Related work

Our work focuses on the task of named entity recognition, on model efficiency and on improving model efficiency. These topics are hardly new to the NLP arena and we will use this section to describe some of the previous work.

2.1 Named Entity Recognition

The most well-known NER task is probably the CoNLL-2003 Task created by Tjong Kim Sang

¹The code for the project is available at <https://github.com/lovhag/distilling-in-swedish>.

and De Meulder (2003). It tests a model on its capacity to recognize words as either names of person (PER), location (LOC), organization (ORG), miscellaneous (MISC) or not an entity (O).

Much work has been done on NER for English, with several models trained on the data, as seen in section 2.2. However, the same cannot be said for other languages. Firstly, there is the issue of obtaining an adequate training, development and test dataset for NER. The largest Swedish dataset which can be used for NER is built on the SUC 3.0 dataset (Ejerhed et al., 1992).

NER data resources have also been developed in other North-Germanic languages and work on this is ongoing. Recently, Hvingelby et al. (2020) created a novel NER dataset for Danish. In the same article, they provide an overview of the available NER datasets for similar languages, such as Swedish and Norwegian. They also train a BERT model for their Danish NER task and obtain an f1 score of 83.76.

2.2 Named Entity Recognition models

When Devlin et al. (2019) tested their BERT model on the downstream task of NER they used the CoNLL-2003 English data and obtained an f1 score of 92.4 with their base model.

One previous state of the art model for NER before BERT, named “CCNN+WLSTM+CRF”, is provided by Yang and Zhang (2018) and Ma and Hovy (2016).² It does not use hand-crafted features or deep contextualized word embeddings.

2.3 Model efficiency

Research that focuses on model efficiency and energy consumption is seemingly on the rise. The most noteworthy contribution within the field of NLP is that of Strubell et al. (2019). In their work, Strubell et al. claim that the NLP field would benefit from reporting training time and sensitivity to hyperparameters for developed models. Additionally, Clark et al. (2020) argue that compute efficiency should be taken in consideration together with downstream performance for representation learning methods. To this end, they report model performance as a function of train FLOPS necessary to reach that performance.

²According to http://nlpprogress.com/english/named_entity_recognition.html.

2.4 Development of more efficient models

Several methods for making models within language processing more efficient have been developed and research on this is ongoing. Seemingly, the methods so far discovered can be categorized into three different types: 1) conditional computation, 2) improving sample efficiency and 3) model compression. Conditional computation is about not using the full network when making inferences, thus reducing the number of computations needed (Shazeer et al., 2017; Fedus et al., 2021). The goal of improving sample efficiency is quite self-explanatory, and may be exemplified by the recent work by (Clark et al., 2020) in which a more effective method for training BERT is proposed. Model compression is the focus of this article and will be further explained in this section.

The objective of model compression is to compress a large model with good performance into a smaller model that still performs on par with the larger model. A “smaller model” is a model which in some way requires less computational power and/or memory. In general, this means that you still need to do some training of the larger model before you can compress it. As such, model compression is beneficial when you want to achieve energy efficiency at deployment. Apart from knowledge distillation, pruning can also be used to this end. For example, after the lottery ticket hypothesis was presented for neural networks by Frankle and Carbin (2018), Chen et al. (2020) presented corresponding work on iterative pruning for BERT models.

Knowledge distillation (KD) is another model compression technique that will be the main focus of this article. The main idea behind the technique is to distill the knowledge from a larger model, a teacher, into a smaller model, a student, by providing the student with the predictions of the teacher (Hinton et al., 2015).

KD can be implemented in different ways during training of the student model. One implementation that was used by Adhikari et al. (2020) is to train the student model to also imitate the predictions of the teacher model through an additional KD term in the loss signal. This KD loss term measures how similar the predictions of the student model $\mathbf{y}^{(s)}$ are to those of the teacher model $\mathbf{y}^{(t)}$, denoted $L_{KD}(\mathbf{y}^{(s)}, \mathbf{y}^{(t)})$. The standard loss for the task, denoted $L_{task}(\mathbf{y}^{(s)})$, is still included in the loss signal. Thus, the training loss during

KD can be described as below.

$$L = L_{\text{task}}(\mathbf{y}^{(s)}) + \lambda L_{\text{KD}}(\mathbf{y}^{(s)}, \mathbf{y}^{(t)}) \quad (1)$$

Here, λ is a tunable hyperparameter used to tune the balance between how much feedback the student model should receive from the objective of the task and how much feedback it should receive from the teacher. With a non-zero λ , the student model is partly trained to imitate the predictions of the teacher model.

KD within the scope of natural language processing can be used in either of two training situations; 1) during pre-training of a model that is intended to be transferable on several downstream tasks and 2) during fine-tuning of a model for a specific downstream task.

Previous work on KD for language models intended to be transferable is that of Sanh et al. (2019) in which a distilled version of BERT (DistilBERT) was created. DistilBERT has 40% fewer model parameters than BERT and is capable of being fine-tuned to perform well on several downstream tasks without requiring as many computations as BERT.

Previous work on KD for a specific downstream task includes that by Adhikari et al. (2020). In their work Adhikari et al. found that generally any model benefits from KD for document classification. They also found that simpler models such as logistic regression models benefit the most with respect to relative improvement in f1 score.

There is also work on trying to understand *why* models benefit from KD. The number of theoretical justifications are few, although some have been found in the recent work by Rahbar et al. (2020). On the other hand, there is more work in the area of empirical explanations. Based on empirical experiments, Yuan et al. (2020) claim that the benefits of KD mainly come from the label smoothing regularization provided by the soft targets of the teacher model, such that even a “bad” teacher can improve the performance of a student model as long as it provides soft targets. Yuan et al. also suggest that an increase in performance that is comparable to that of KD can be obtained by using “self-training” or a manually designed regularization term, without the need of a teacher model.

3 Swedish NER dataset

We use the manual NER annotations based on the SUC 3.0 dataset (Ejerhed et al., 1992) for our

SUC 3.0	CoNLL-2003
person	PER
animal	PER
myth	PER
place	LOC
institution	ORG
product	MISC
work	MISC
event	MISC
other	MISC

Table 1: The mapping used to convert SUC 3.0 entity types to the same as those of the CoNLL-2003 data.

Resource	SUC 3.0
#tokens	1,166,593
#entity tokens	47,310
%entities	4.06

Table 2: Some general features of the SUC 3.0 NER dataset in Swedish. The number of entity tokens measures the number of tokens that make up the named entities. The percentage of entities is the number of tokens that make up entities divided by the total number of tokens in the dataset.

Swedish NER task. Before training, we reshape the data to a more suitable format for our task.

Firstly, the manual annotations in the SUC 3.0 data contain annotations for the entities person, animal, myth (for example “God”), place, institution, product, work, event and other. These entity categories are not found in NER datasets for other languages. In order to make better comparisons to other languages, we map the entity types in the dataset to the same types as those that can be found in the CoNLL-2003 data, as described by Table 1. We also represent the data in the IOB2 format (Tjong Kim Sang and Veenstra, 1999) and split it into 70%/10%/20% for the train/validation/test data. The splits were made with random sampling without regard to text source.

Tables 2 and 3 list some of the features of the reshaped dataset. From these tables, we can observe that the Swedish dataset is about three times larger than the CoNLL-2003 dataset, while the latter has a higher density of entities. It is worth remarking that while the English dataset was developed for NER, the SUC dataset was originally compiled for the purpose of part-of-speech tagging, with the entity annotation added later. Additionally, we can

Resource		LOC	MISC	ORG	PER	#examples
SUC 3.0	train	6,705	4,551	6,005	16,030	51,971
	dev	955	549	885	2,135	7,351
	test	1,857	1,402	1,574	4,662	14,923
	total	9,517	6,502	8,464	22,827	74,245

Table 3: The distribution of the named entities of the Swedish NER dataset.

observe from Table 3 that the Swedish dataset has quite an unbalanced entity distribution.

4 Method

The goal of this work is contribute to better model efficiency within NLP by investigating the effect of KD on different NER models in Swedish. To this end, we utilize the method for KD as presented in Section 4.1 and investigate the NER models seen in Section 4.2. The efficiency of our models is then measured as described in Section 4.3.

4.1 Application-targeted KD

The general form of the KD objective was previously introduced in Equation (1). We let the KD loss term $L_{\text{KD}}(x)$ for one batch be given by the Kullback–Leibler divergence as shown in Equation (2), similarly to what was done by Adhikari et al. (2020).

$$L_{\text{KD}}(\mathbf{y}^{(s)}, \mathbf{y}^{(t)}, \mathbf{w}) = \sum_n \sum_{l: \mathbf{w}_{n,l} \neq \text{“PAD”}} \sum_k \frac{\mathbf{y}_{n,l,k}^{(t)}}{N} \left(\log \frac{\mathbf{y}_{n,l,k}^{(t)}}{\mathbf{y}_{n,l,k}^{(s)}} \right) \quad (2)$$

$\mathbf{y}^{(s)}$ and $\mathbf{y}^{(t)}$ are the respective label probabilities of student and teacher model for each token in each batch example. The sum indices n , l , k denote the batch index, token index and label index. So $1 \leq n \leq 32$, $1 \leq l \leq 128$ and $1 \leq k \leq 9$ in the case of our work. N is the batch size and $\mathbf{w}_{n,l}$ denotes the token at position l in the sequence with index n in the batch.

The objective of the KL divergence is to measure the difference between the student model label probabilities and the teacher model label probabilities. It is only zero if the probabilities are identical. Neither the cross-entropy loss nor the Kullback–Leibler divergence were evaluated for padding tokens.

Another important variable for the KD is λ . This was set by studying the sizes of the two loss terms and making sure that they contributed

with feedback of roughly equal magnitude, as this seemingly generated the best KD results.

Moreover, data augmentation has successfully been used for improving the performance of KD (Hinton et al., 2015). Results by Ba and Caruana (2014a) indicate that the more data, the more for the student model to learn on from the teacher model. To this end, Adhikari et al. (2020) used data augmentation during KD. However, we find it meaningful to investigate the benefits of KD before the usage of data augmentation, and will not use it in this work.

4.2 Models

All of the evaluated models and their parameters are listed in Table 4. The models were chosen with the objective of investigating the effect of KD for simpler as well as more complex models on the NER task, similarly to what was done by Adhikari et al. (2020). However, we did not include quite as simple models as those evaluated by Adhikari et al., as our sequence tagging task of NER requires a sequential model output.

Common for all models except for the Char-CNNWordLSTM model is that their input is formatted by a Swedish BERT tokenizer with a vocabulary size of 50,325. As such, each embedding layer of the models expects word pieces as input and covers a vocabulary of the same size as BERT. Additionally, each training example is truncated or padded to a sequence length of 128 word-pieces and the label for an entity consisting of several word pieces is given by the label generated for the first wordpiece, similarly to the approach by Devlin et al. (2019).

The BERT model was developed with support from the Huggingface Transformers software by Wolf et al. (2020). A linear classification layer was added on top of the pre-trained Swedish base BERT model by Malmsten et al. (2020) to create a BERT model for NER in Swedish. This model was fine-tuned for 3 epochs on the Swedish NER data. A cross-entropy loss was used and all layers of the model were fine-tuned during training. This

Model name	#parameters	% in emb	infer FLOPS	% of BERT FLOPS	infer time [s]
BERT	124,107,273	31.14	2.9e10	100	0.287
Window	6,445,065	99.94	8.8e5	3e-3	0.000254
Window-B	38,670,345	99.95	5.3e6	2e-2	0.000718
LSTM-128	6,708,105	96.03	6.9e7	2e-1	0.009278
LSTM-128-B	39,571,465	97.67	2.4e8	8e-1	0.011056
LSTM-256	13,940,489	92.42	2.7e8	9e-1	0.015666
LSTM-256-B	40,755,465	94.83	5.4e8	2	0.020334
LSTM-256-2-B	42,332,425	91.30	9.4e8	3	0.035662
LSTM-256-2-drop-B	42,332,425	91.30	9.4e8	3	0.037428
CharCNNWordLSTM	27,002,212	98.75	1.0e8	4e-1	0.009930

Table 4: The number of model parameters for all models investigated. We also indicate how many percentages of the parameters are found in the word embedding layer. The infer FLOPS correspond to one forward pass of an example. The infer time is the inference time of the model for one example.

model also served as the teacher during KD training of the other models in Table 4, such that $\mathbf{y}^{(t)}$ in Equation (2) is given by the predictions of this model.

The Window model is a straightforward implementation of a window-based sequence labeling model with a window size of 3. This window size was found to be the best after some preliminary tuning. Furthermore, the model has an initial embedding layer with dimension (50325, 128) and a final fully connected top layer which predicts for the nine available labels.

The LSTM-128 model is a straightforward implementation of an LSTM model with an initial embedding layer with dimension (50325, 128), a hidden bidirectional LSTM layer with size 128 and a final fully connected top layer for the labels. The same applies for the LSTM-256 model, with the exception that the LSTM layer of this model has a size of 256 and that the embedding dimension is 256.

The LSTM-256-2 model has the same architecture as the LSTM-256 except for that it utilizes two bidirectional LSTM layers instead of one.

The LSTM-256-2-drop model has the same architecture as the LSTM-256-2 model except for that it has a word dropout probability of 0.2 and a dropout layer with a dropout probability of 0.2 on the output of the first LSTM layer. This model was chosen to investigate the effect of KD on a more regularized model.

The -B extension denotes that the same model architecture is used, but with the pre-trained word piece embedding layer of size 50325×768 from the BERT model. For these -B models the em-

bedding layer is frozen during training. Consequently, this increases the number of parameters for the models, while it is somewhat mitigated by the fact that the embedding layer does not need to be tuned.

The CharCNNWordLSTM model was chosen with the intention of investigating the effect of KD on a state-of-the-art model for NER which does not utilize deep contextual word representations. The architecture of this model is a CharCNN+WordLSTM structure, the same as that of Yang and Zhang (2018) and Ma and Hovy (2016), with the exception that we do not include the conditional random field (CRF) layer in our CharCNNWordLSTM model. Similarly to the work by Yang and Zhang (2018) and Ma and Hovy (2016) we use pre-trained word embeddings in the model. These are given by a Word2Vec model trained on a Swedish corpus.³ The Swedish embeddings have a word vocabulary of size 104,162 and an embedding size of 256. To make the KD from BERT feasible, the input data to the CharCNNWordLSTM model was potentially truncated to less than 128 words, since the output of it needed to be of the same shape as that of BERT which was given an input of maximum 128 word pieces. Additionally, this model is regularized with dropout and weight decay during training. This model and the LSTM-256-2-drop model are the only models with regularization mechanisms, such as dropout.

Common for all models is that none of them employ a final CRF layer. Models used for se-

³The corpus consists of approximately 10e9 words from a mix of corpora distributed by Språkbanken, <https://spraakbanken.gu.se/resurser>.

quence tagging usually show an improvement in performance if they have a final CRF layer which takes regard to sequential dependencies in the predictions. We chose to not use a CRF layer with the purpose of faster training and a simpler implementation of the KD.

All non-BERT models are trained and evaluated both with and without KD on a GeForce GTX TITAN X GPU. Every model was trained until it showed no further increase in f1 score. The results reported for the models in Tables 5 and 6 are the test scores for the model checkpoint with the best f1 score on the validation data. The number of epochs for the model in the table is then given by the number of train epochs required to reach this best checkpoint.

4.3 Method for measuring model efficiency

To evaluate the method of KD with respect to efficiency we measure the inference time, number of parameters as well as training and inference FLOPS required by each model investigated, as seen in Tables 4 to 6.

We use the Python package `thop` to estimate the number of FLOPS required for one forward-pass of all models in Table 4 except for BERT. These numbers are reported as “infer FLOPS” in the table. The number of FLOPS are calculated for the forward-pass of one data example with a sequence length of 128. We choose a character length of 15 for the forward-pass example in the case of the CharCNNWordLSTM model which also separates the characters of each word. We do not include the FLOPS required by the embedding layer in these calculations since we deem this number to be negligible in comparison with the FLOPS required by the other parts of the models.

To estimate the number of FLOPS required for training we then use Equation (3).⁴ In the equation, n_{infer} denotes the number of FLOPS required for one forward pass and n_{examples} denotes the number of examples the model was trained on. The number of training FLOPS is reported as “FLOPS” in Tables 5 and 6.

$$n_{\text{FLOPS}} = n_{\text{infer}} \cdot 3 \cdot n_{\text{examples}} \quad (3)$$

To calculate the number of FLOPS required for one forward pass in the BERT model, which is

⁴There is a blog post by OpenAI which explains a method for calculating model training FLOPS, see <https://openai.com/blog/ai-and-compute/>.

a standard BERT-base model, we use the information given by Clark et al. (2020). The pre-train FLOPS required for BERT are then given by estimating the training parameters of the BERT training method as described by Malmsten et al. (2020) and using Equation (3) with the forward pass FLOPS previously obtained.

To calculate the inference time, denoted “infer time” in Table 4, we use the same data example as was used for calculating the number of infer FLOPS for the models. We then make the model predict for this example 100 times and estimate the average of the inference time required for each prediction iteration as the inference time of the model. These time calculations were done on a 2.3 GHz Quad-Core Intel Core i7 CPU.

5 Results and Discussion

The model scores on the Swedish NER test data are split into Tables 5 and 6. The results in the former table are of the simpler models that were not regularized, while the results in the latter are of the models that were regularized.

We split the analysis of the results with respect to our aspects of interest. Consequently, we start off with a general analysis of the model results for Swedish NER, after which we examine the effect of KD on model scores and then study the effect of KD on model efficiency.

5.1 General analysis of the Swedish NER model results

Firstly, the BERT model has the highest f1 score for the Swedish NER task. This also comes with the highest computational cost and the longest inference time, which is ten times longer than that of the second most slow model. This is not surprising, as the current trend within NLP is that better models require more resources.

Moreover, the f1 score of the BERT model is approximately two percentage units lower than that of BERT on the English NER dataset. This could be due to the difference between the datasets, different fine-tuning procedures, and/or to the different pre-training processes of the BERT models. Nonetheless, it is not entirely unexpected that the models we investigate may perform worse for the Swedish language than for the English.

Model	P	R	f1	epochs	FLOPS
Window	0.667 ± 0.005	0.707 ± 0.007	0.686 ± 0.004	18 ± 5	$2.4e12$
KD	0.681 ± 0.006	0.705 ± 0.003	0.693 ± 0.004	18 ± 2	$2.4e12$
B	0.731 ± 0.000	0.721 ± 0.002	0.726 ± 0.001	24 ± 4	$2.0e13$
B-KD	0.726 ± 0.001	0.712 ± 0.002	0.719 ± 0.000	21 ± 2	$1.8e13$
LSTM-128	0.720 ± 0.004	0.717 ± 0.004	0.719 ± 0.003	60 ± 9	$6.5e14$
KD	0.758 ± 0.006	0.736 ± 0.005	0.747 ± 0.005	66 ± 11	$7.1e14$
B	0.802 ± 0.004	0.808 ± 0.005	0.805 ± 0.004	44 ± 23	$1.7e15$
B-KD	0.823 ± 0.003	0.822 ± 0.004	0.823 ± 0.003	60 ± 12	$2.2e15$
LSTM-256	0.743 ± 0.007	0.729 ± 0.008	0.735 ± 0.006	46 ± 24	$1.9e15$
KD	0.784 ± 0.005	0.747 ± 0.003	0.765 ± 0.003	66 ± 15	$2.8e15$
B	0.807 ± 0.010	0.815 ± 0.004	0.811 ± 0.006	54 ± 12	$4.6e15$
B-KD	0.829 ± 0.006	0.826 ± 0.002	0.828 ± 0.003	66 ± 21	$5.5e15$
LSTM-256-2					
B	0.830 ± 0.007	0.831 ± 0.003	0.831 ± 0.005	61 ± 21	$8.9e15$
B-KD	0.849 ± 0.004	0.845 ± 0.004	0.847 ± 0.004	78 ± 15	$1.1e16$

Table 5: The scores on the test data for all of the evaluated models that are not regularized.

Model	P	R	f1	epochs	FLOPS
BERT	0.892	0.897	0.895	3	$1.4e16$ ($9.1e19$)
LSTM-256-2-drop-B	0.844 ± 0.006	0.832 ± 0.002	0.838 ± 0.002	39 ± 9	$5.7e15$
KD	0.847 ± 0.004	0.833 ± 0.006	0.840 ± 0.002	25 ± 10	$3.6e15$
CharCNNWordLSTM	0.843 ± 0.002	0.822 ± 0.004	0.836 ± 0.008	90 ± 11	$1.4e15$
KD	0.842 ± 0.005	0.824 ± 0.003	0.833 ± 0.003	97 ± 2	$1.5e15$

Table 6: The scores on the test data for all of the evaluated models implemented with regularization. Models trained with knowledge distillation are marked with “KD”. “P” denotes precision and “R” recall. Epochs, time and mean number of FLOPS required to reach best evaluation performance during training are also displayed. FLOPS values in parentheses denote number of FLOPS required during pre-training.

5.2 The effect of KD on model scores

For the one-layer LSTM models without BERT embeddings the f1 score increases with approximately 3 units when using KD training. With BERT embeddings, these models also benefit some from KD. Seemingly, the LSTM models improve primarily in precision when KD is applied.

Additionally, it appears as though the LSTM models benefit more from KD than the simpler Window model. The Window model without BERT embeddings displays an increase in precision with KD, while the same model with BERT embeddings even decreases in performance with KD. This contradicts previous results on KD by e.g. Adhikari et al. (2020), where it was found that simpler models have the most to benefit from KD. A potential reason for this could be that the model architecture was not expressive enough to benefit from KD.

Moreover, the LSTM-256-2-drop-B model per-

forms better than its counterpart LSTM-256-2-B when no KD is applied. However, when KD is applied, the LSTM-256-2-B-KD model surpasses the LSTM-256-2-drop-B-KD model in f1 score as it seemingly benefits more from KD.

The models Window-B, LSTM-256-2-drop-B and CharCNNWordLSTM that do not clearly benefit in f1 score from KD have in common that they are either quite small or regularized. Revisiting the idea of Yuan et al. (2020), one possible reason for this is that KD provides regularization and that a model that does not need regularization consequently will not benefit in performance from KD.

5.3 The effect of KD on model efficiency

The three non-BERT models with the best f1 scores in descending order are given by the LSTM-256-2-B-KD, LSTM-256-2-drop-B-KD and the CharCNNWordLSTM models. The LSTM models are slightly better than the CharCNNWordLSTM model, although this comes with

the price of requiring approximately 4 to 10 times more FLOPS for training and an inference time that is approximately 4 times longer. The LSTM models also rely on the existence of a pre-trained BERT model, which requires approximately $9.1e19$ FLOPS. While the Char-CNNWordLSTM model also relies on pre-trained word embeddings, these do not require as many FLOPS.

The best non-BERT model is the LSTM-256-2-B-KD with an f1 score of 0.847. It is approximately 5 units worse than the BERT model, while it requires approximately the same number of training FLOPS (BERT pre-training not included) and only 3% of the number of inference FLOPS required by BERT. Clearly, it is more efficient at deployment, while the question remains as to whether it has a performance good enough for deployment.

The second best non-BERT model is the LSTM-256-2-drop-B-KD model. While it did not clearly benefit in f1 score from KD, it seemingly benefited with respect to the number of required training FLOPS, as the number of training FLOPS of the model decreased with approximately 40%. In every other case, the general model behavior with KD applied is that both the number of training FLOPS and the f1 score increase.

Clearly, every trained model that utilized and benefited from KD is more performance efficient than its non-distilled version when making inferences for new data, as the model improves in f1 score while the number of computations for inference is the same as before KD. However, the cost of training such a model is higher, mainly due to the need of a trained teacher model. The question is whether the gain in deployment efficiency is worth the additional effort. One way to reason about this is through basic arguments of when such an “investment” would reach a break-even point, similarly to how e.g. solar panels are judged based on how many years they would need to be used to repay the energy required to produce them. For example, if we are to develop a model that we know will be run several times during deployment, the use of KD could enable the use of a smaller model without loss of performance, thus reducing the computational cost required during deployment, weighting up for the extra cost of training it with KD. One such model that has been developed is DistilBERT (Sanh et al., 2019), which only last

month was downloaded 1,544,446 times from the Huggingface model library.⁵

Apart from reasoning about model efficiency, we can also reason about when an increase in f1 score is worth the associated computational cost. Since the general trend is that we obtain better models if we allow for an increase in computational cost, the question is how much we are willing to pay for one unit of f1 score. In this case we also have to take into account that the computational cost of one f1 score unit increases with f1 score, as it is harder to increase the performance in the region of e.g. 0.9 than it is in the region of 0.6. Ethayarajh and Jurafsky (2020) propose a way to handle this by using an utility function that takes regard to performance as well as practical concerns, such as model size and inference latency. It may be appropriate to investigate the effect of KD in the eye of such an utility function.

6 Conclusion and Future work

Our work indicates that different models may differ in whether they benefit from KD. Thus, we cannot make the assumption that KD should benefit the performance of every model. Adding to the question of *why* some models seem to benefit from KD, we may also ask ourselves *in which situations* the soft targets of a teacher model may benefit a student model.

We observe three different situations for which it is worth to further investigate the effect of KD; 1) when the student model is in need of regularization, 2) when we want more data for the student model to train on and 3) when the data for training is of poor quality. The two latter situations have not been covered in this work, while they have been mentioned by other researchers (Ba and Caruana, 2014b). The former situation has already been observed by Hinton et al. (2015), and we have found additional support for it in our work. For this situation we may further investigate how KD works in combination with existing regularization techniques and whether it is a better such technique.

From our work we can also conclude that KD may provide us with more efficient models at deployment, while the cost of training these models is high due to the need of a trained teacher. This prompts us to reason about when KD is worth the effort, with regard to how we value an increase in

⁵For the uncased base version of DistilBERT.

f1 score in terms of computational costs. We also reason about situations when KD may be a good investment for models that will be used heavily during deployment. To fully measure the benefits of KD with respect to model efficiency we conclude that we need to investigate better tools for judging these trade-offs and different deployment situations.

Future work may also investigate other types of KD, such as extracting more layers than the embedding layer from the teacher model and providing teacher signals to more layers of the student model. Potentially, these KD variations could further improve the performance of a model without requiring more computational costs.

Moreover, it still remains to investigate the benefits of data augmentation for the student models during KD. The question is whether we could attain even better KD results with this approach. This could also be taken one step further to the region of completely unsupervised training on unlabeled data, merely by providing the student model with the labels generated by the teacher.

Lastly we can conclude that, unsurprisingly, KD works for the Swedish language as well. One interesting next step which may benefit the Swedish industry would be to develop a Swedish DistilBERT.

Acknowledgements

We would like to thank Tobias Norlund for his feedback during the writing process of this article. We would also like to thank the anonymous reviewers for their feedback and valuable input.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

- Ashutosh Adhikari, Achyudh Ram, Raphael Tang, William L. Hamilton, and Jimmy Lin. 2020. Exploring the limits of simple learners in knowledge distillation for document classification with DocBERT. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 72–77, Online. Association for Computational Linguistics.
- Jimmy Ba and Rich Caruana. 2014a. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, volume 27, pages 2654–2662. Curran Associates, Inc.
- Jimmy Ba and Rich Caruana. 2014b. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, volume 27, pages 2654–2662. Curran Associates, Inc.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.
- Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The lottery ticket hypothesis for pre-trained BERT networks. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada.
- Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota.
- Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. 1992. The linguistic annotation system of the Stockholm-Umeå corpus project – description and guidelines. Technical report, Department of Linguistics, Umeå University.
- Kawin Ethayarajh and Dan Jurafsky. 2020. Utility is in the eye of the user: A critique of nlp leaderboard design. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4846–4853.
- William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*.
- Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Training pruned neural networks. *CoRR*, abs/1803.03635.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Rasmus Hvingelby, Amalie Brogaard Pauli, Maria Barrett, Christina Rosted, Lasse Malm Lidegaard, and Anders Søgaard. 2020. DaNE: A named entity resource for Danish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4597–4604, Marseille, France. European Language Resources Association.
- Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany.
- Martin Malmsten, Love Börjeson, and Chris Haf-fenden. 2020. Playing with words at the National Library of Sweden – making a Swedish BERT.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Arman Rahbar, Ashkan Panahi, Chiranjib Bhattacharyya, Devdatt Dubhashi, and Morteza Haghiri Chehreghani. 2020. On the unreasonable effectiveness of knowledge distillation: Analysis in the kernel regime. *arXiv preprint arXiv:2003.13438*.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Emma Strubell, Ananya Ganesh, and Andrew McCalum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

- Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 173–179, Bergen, Norway.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Jie Yang and Yue Zhang. 2018. NCRF++: An open-source neural sequence labeling toolkit. In *Proceedings of ACL 2018, System Demonstrations*, pages 74–79, Melbourne, Australia.
- Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jishi Feng. 2020. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911.

Fine-grained Named Entity Annotation for Finnish

Jouni Luoma, Li-Hsin Chang, Filip Ginter, Sampo Pyysalo

TurkuNLP group

Department of Computing,

Faculty of Technology

University of Turku, Finland

{jouni.a.luoma, lhchan, figint, sampo.pyysalo}@utu.fi

Abstract

We introduce a corpus with fine-grained named entity annotation for Finnish, following the OntoNotes guidelines to create a resource that is cross-lingually compatible with existing resources for other languages. We combine and extend two NER corpora recently introduced for Finnish and revise their custom annotation scheme through a combination of automatic and manual processing steps. The resulting corpus consists of nearly 500,000 tokens annotated for over 50,000 mentions categorized into 18 name and numeric entity types. We evaluate this resource and demonstrate its compatibility with the English OntoNotes annotations by training state-of-the-art mono-, bi-, and multilingual deep learning models, finding both that the corpus allows highly accurate tagging at 93% F-score and that a comparable level of performance can be achieved by a bilingual Finnish-English NER model.¹

1 Introduction

Named Entity Recognition (NER), the identification and typing of text spans referring to entities such as people and organizations in text, is a key task in natural language processing. State of the art NER approaches apply supervised machine learning methods trained on corpora that have been manually annotated for mentions of entity names of interest. While extensive corpora with fine-grained NER annotation have long been available for high-resource languages such as English, NER for many lesser-resourced languages has been limited by smaller, lower-coverage corpora with comparatively coarse annotation.

¹The corpus is available under an open license from <https://github.com/TurkuNLP/turku-one>

A degree of language independence has long been a central goal in NER research. One notable example are the CoNLL shared tasks on Language-Independent Named Entity Recognition in 2002 and 2003 (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003). The Spanish, Dutch, English and German datasets introduced in these shared tasks were all annotated for the same types of entity mentions – persons, organizations, locations, and miscellaneous – and the datasets still remain key benchmarks for evaluating NER methods today (e.g. (Devlin et al., 2019)). Nevertheless, until recently most NER methods aimed for language independence only in that they supported training on corpora of more than one language, resulting in multiple separate monolingual models.

In recent years, advances in deep learning have made it possible to create multilingual language models that achieve competitive levels of performance when trained and applied on texts representing more than one language (e.g. Kondratyuk and Straka (2019)). One notable model is the multilingual version of the influential BERT model (Devlin et al., 2019), mBERT, trained on more than 100 languages. mBERT performs well on zero-shot cross-lingual transfer experiments, including NER experiments (Wu and Dredze, 2019). Moon et al. (2019) propose an mBERT-based model trained simultaneously on multiple languages. Training and validating on the OntoNotes v5.0 corpus (see Section 2.3) and the CoNLL datasets, they show that multilingual models outperform models trained on one single language and have cross-lingual zero-shot ability. The zero-shot cross-lingual transfer ability of mBERT also spikes interest in the study of multilingual representations, both on mBERT (Pires et al., 2019; K et al., 2020), and on multilingual encoders in general (Ravishankar et al., 2019; Zhao et al., 2020; Choenni and Shutova, 2020).

Corpus	Language	Tokens	Entities	Domain(s)
OntoNotes	English	2.0M	162K	News, magazines, conversation
FiNER	Finnish	290K	29K	Technology news, Wikipedia
Turku NER	Finnish	200K	11K	News, magazines, blogs, Wikipedia, speech, fiction, etc.

Table 1: Corpus features and statistics. OntoNotes token count only includes sections of the corpus annotated for name mentions. Entity counts include also non-name types such as DATE.

In this paper, we aim to assess and realize the potential benefits from cross- and multi-lingual NER for Finnish, a lesser-resourced language that currently lacks NER resources annotated compatibly with larger similar resources in other languages. Recently, two NER corpora were introduced for Finnish: FiNER (Ruokolainen et al., 2019), focusing on the technology news domain, and the Turku NER corpus (Luoma et al., 2020), covering 10 different text domains. The two corpora are both annotated in the same custom variant of the CoNLL’02 and ’03 scheme, making them mutually compatible, but incompatible with resources existing in other languages. This incompatibility has so far made it impossible to directly evaluate the performance of cross- and multi-lingually trained NER methods on manually annotated Finnish resources. To solve this incompatibility issue, we combine and extend these two corpora and adjust the annotations to follow the OntoNotes scheme. The resulting corpus has close to 500,000 tokens annotated for over 50,000 mentions assigned to the 18 OntoNotes name and numeric entity types. We show that our OntoNotes Finnish NER corpus is compatible with the English OntoNotes annotations through training state-of-the-art bi- and multilingual NER models on the combination of these two resources.

2 Data

In the following, we introduce the corpora used in this study, additional text sources for the new corpus, and the pre-trained models used in our experiments. The properties and key statistics of the corpora are presented in Table 1.

2.1 FiNER corpus

FiNER (Ruokolainen et al., 2019) is a Finnish NER corpus consisting mainly of texts from the Finnish technology news source Digitoday, with an additional test set of Wikipedia documents used to assess cross-domain performance of methods trained on the FiNER training section.

FiNER is annotated for mentions of dates (type DATE) and five entity types: person (PER), organization (ORG), location (LOC), product (PRO) and event (EVENT). Of these, PER, ORG and LOC are broadly compatible with the CoNLL types of the same names. The original corpus includes a small number of nested annotations (under 5% of the total) that were excluded in our work.

2.2 Turku NER corpus

The Turku NER corpus (Luoma et al., 2020) is a Finnish NER corpus initially created on the basis of the Universal Dependencies (Nivre et al., 2016) representation of the manually annotated Turku Dependency Treebank (TDT) (Haverinen et al., 2014; Pyysalo et al., 2015), a multi-domain corpus spanning ten different genres.

The Turku NER annotation follows the types and annotation guidelines of the FiNER corpus. An evaluation by Luoma et al. (2020) demonstrated the compatibility of the two Finnish NER corpora by showing that models trained on the simple concatenation of the two corpora outperformed ones trained on either resource in isolation.

2.3 OntoNotes corpus

OntoNotes (Hovy et al., 2006; Weischedel et al., 2013) is a large, multilingual (English, Chinese, and Arabic), multi-genre corpus annotated with several layers covering text structure as well as shallow semantics. In this work, we focus exclusively on the OntoNotes English language NER annotation and refer to this part of the data simply as OntoNotes for brevity. Specifically, we use the NER annotations of the OntoNotes v5.0 release (Weischedel et al., 2013), cast into CoNLL-like format by Pradhan et al. (2013).² Sections of the corpus lacking NER annotation (such as the Old and New Testament texts) are excluded.

The OntoNotes NER annotation uses a superset of the ACE entity annotation representation (LDC,

²<https://github.com/ontonotes/conll-formatted-ontonotes-5.0>

Type	Description	Examples
PERSON	People, including fictional	Keijo Virtanen, Obama
NORP	Nationalities or religious or political groups	suomalainen, kristitty
FAC	Buildings, airports, highways, bridges, etc.	Turun linna, LHC
ORG	Companies, agencies, institutions, etc.	Nokia, EU
GPE	Countries, cities, states	Suomi, Venäjä
LOC	Non-GPE locations, mountains, bodies of water	Välimeri, Ararat
PRODUCT	Objects, vehicles, foods, etc. (Not services.)	Oltermanni, iPhone
EVENT	Named hurricanes, wars, sports events, etc.	toinen maailmansota, CES
WORK_OF_ART	Titles of books, songs, etc.	Raamattu, Kid A
LAW	Named documents made into laws	rikoslaki, Obamacare
LANGUAGE	Any named language	suomi, englantia, C++
DATE	Absolute or relative dates or periods	viime vuonna, 1995
TIME	Times smaller than a day	yö, viisi sekuntia
PERCENT	Percentage, including “%”	seitsemän prosenttia, 12%
MONEY	Monetary values, including unit	sata euroa, 500 dollaria
QUANTITY	Measurements, as of weight or distance	kilometri, 5,1 GHz
ORDINAL	“first”, “second”	ensimmäinen, 1.
CARDINAL	Numerals that do not fall under another type	yksi, kaksi, 10

Table 2: OntoNotes name annotation types. Adapted from Weischedel et al. (2013).

Model	Language(s)	Vocab. size	Reference
BERT (original)	English	30K	Devlin et al. (2019)
FinBERT	Finnish	50K	Virtanen et al. (2019)
mBERT	104 languages	120K	Devlin et al. (2019)
biBERT	Finnish and English	80K	Chang et al. (2020)

Table 3: Pre-trained models. Cased base variants of all models are used.

2008), applying the 18 types summarized in Table 2. We note that while OntoNotes PERSON, EVENT and DATE largely correspond one-to-one to types annotated in the Finnish NER corpora, the great majority of the types either require a more complex mapping or need to be annotated without support from existing data to create OntoNotes annotation for Finnish.

2.4 Additional texts

During annotation, we noted that the FiNER and Turku NER corpora contained relatively few mentions of laws, which could potentially lead to methods trained on the combined revised corpus performing poorly on the recognition of LAW entity mentions. To address this issue, we augmented the combined texts of the two corpora with a random selection of 60 current acts and decrees of Finnish Acts of Parliament,³ totaling approximately 24K tokens.

³Available from <https://finlex.fi/fi/laki/ajantasa/>

2.5 Pre-trained models

We perform NER tagging experiments by fine-tuning monolingual and multilingual BERT models. Specifically, for monolingual models, we tested English and Finnish (FinBERT) models, and for multilingual models, we tested the mBERT model trained on 104 languages, and a bilingual model trained on only English and Finnish (biBERT). Devlin et al. (2019) trained the original English BERT on the BooksCorpus (Zhu et al., 2015) and English Wikipedia. FinBERT is trained on an internet crawl, news, as well as online forum discussions (Virtanen et al., 2019). The bilingual BERT is trained on English Wikipedia and a reconstructed BooksCorpus, as well as the data used to train FinBERT (Chang et al., 2020). The multilingual BERT is trained on the Wikipedia dump for languages with the largest Wikipedias. The pre-trained models and their key statistics are summarized in Table 3.

We note that while a number of variations and improvements to the pre-training of transformer-

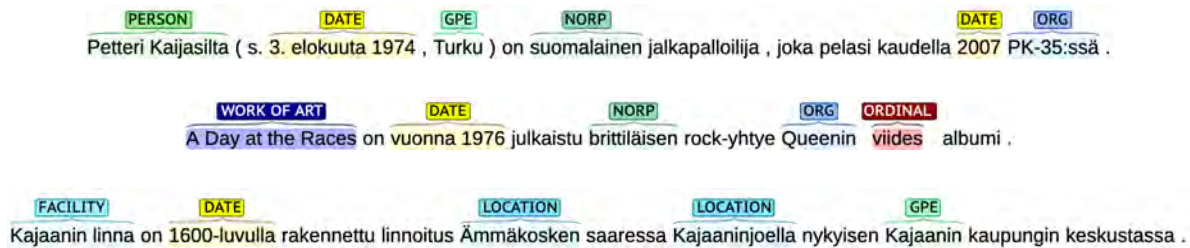


Figure 1: Example annotations

based deep language models have been proposed since the introduction of BERT (e.g. Conneau et al. (2019); Xue et al. (2020)), BERT remains by far the most popular choice for training monolingual deep language models and an important benchmark for evaluating methods for tasks such as NER. As the focus of our evaluation is more on assessing the quality and compatibility of corpora through the application of comparable models rather than optimizing absolute performance, we have here opted to use exclusively BERT models. For the same reason, we only consider BERT base models instead of a mix of base and large models.

3 Annotation

We next summarize the primary steps performed to revise and extend the annotation of the two source corpora to conform with the OntoNotes NER guidelines (Weischedel et al., 2013). Figure 1 shows visualizations of the annotation for selected sentences.

Trivial mappings Of the mentions annotated in the existing Finnish NER corpora, effectively all annotations with the type `PER` are valid OntoNotes `PERSON` annotations. Similarly, most `EVENT` and `DATE` annotations were valid as-is as OntoNotes annotations of the same names. These annotations were carried over into the initial revised data, changing only the type name when required.

Conditional mappings By contrast to the types allowing trivial mapping from existing to revised annotation, `LOC`, `ORG` and `PRO` required more complex mapping rules. For example, the existing annotations mark both geo-political entities (GPEs) and other locations with the type `LOC` without distinguishing between the two. To create OntoNotes-compatible annotation, source `LOC` annotations were mapped to either `LOC` or `GPE` annotations on the basis of the annotated

text using manually created rules. For example, `Suomi/LOC` (“Finland”) was mapped to `Suomi/GPE` and `Välimeri/LOC` (“Mediterranean”) to `Välimeri/LOC`. Similar rules were implemented to distinguish e.g. `FAC` from `ORG` and `LOC` as well as `WORK_OF_ART` and `LAW` from `PRO`.

Dictionary-based tagging Not all mentions in scope of the OntoNotes annotation guidelines are in scope of the FiNER annotation guidelines applied to mark the previously introduced Finnish NER corpora. In addition to most OntoNotes numeric types (see below), in particular nationalities, religious and political groups (`NORP` in OntoNotes) and languages (`LANGUAGE`) were not annotated in the source corpora. To create initial OntoNotes annotation for these semi-closed categories of mentions, we performed dictionary-based tagging using lists compiled from sources such as Wikipedia and manually translated OntoNotes English terms tagged with the relevant types.⁴

Numeric types To annotate OntoNotes numeric types (`CARDINAL`, `ORDINAL`, etc.) in the Turku NER corpus section of the data, we mapped the manual part-of-speech and feature annotation of the source corpus (TDT) to initial annotations that were then manually revised to identify the more specific types such as `PERCENT`, `QUANTITY` and `MONEY` based on context. For the FiNER texts, annotation for these types followed a similar process with the exception that automatic part-of-speech and feature annotation created by the Turku neural parser (Kanerva et al., 2018) was used as a starting point as no manual syntactic annotation was available for the texts.

Fine-grained tokenization The FiNER annotation guidelines specify that annotated name men-

⁴The accuracy of this initial dictionary-based tagging step was not evaluated separately.

Language	Model	Train data	Development data	Test data
Finnish	FinBERT	Finnish	Finnish	Finnish
Finnish	mBERT	Combined (Fi+En)	Finnish	Finnish
Finnish	biBERT	Combined (Fi+En)	Finnish	Finnish
English	BERT	English	English	English
English	mBERT	Combined (Fi+En)	English	English
English	biBERT	Combined (Fi+En)	English	English

Table 4: Combinations of models, training and evaluation data included in the experiments.

tions must start and end on the boundaries of syntactic words. As hyphenated compound words that include names as part, such as *Suomi-fani* (“fan of Finland”), are comparatively common in Finnish, the FiNER guidelines have a somewhat complex set of rules for the annotation of such compound words (we refer to Ruokolainen et al. (2019) and the relevant guidelines for details). In the revised corpus, we chose to apply a fine-grained tokenization where punctuation characters (including hyphens) are separate tokens, eliminating most of the issues with names as part of hyphenated compounds. To map FiNER-style annotation to the fine-grained version, we wrote a custom tool using regular expressions and manually compiled white- and blacklists of suffixes that can and cannot be dropped from name mention spans.⁵

Semi-automatic and manual revision After initial automatic revisions, a series of semi-automatic and manual revision rounds were performed using the BRAT annotation tool (Stenetorp et al., 2012). In particular, the consistency of mention annotation and typing was checked using the search functionality of the tool⁶ and all cases where a string was inconsistently marked or typed were revisited and manually corrected when in error. Additionally, the automatically created pre-annotation for the newly added text (Section 2.4) was revised and corrected in a full, manual annotation pass. All manual revisions of the data were performed by a single annotator familiar with the corpora as well as the FiNER and OntoNotes guidelines. While the single-annotator setting regrettably precludes us from reporting inter-annotator agreement, our monolingual and cross-lingual results below suggest that the consistency of the annotation has not decreased from that of the source corpora.

⁵The implementation is available from <https://github.com/spyysalo/finer-postprocessing>

⁶`search.py -cm` and `-ct` options.

4 Methods

We next present the applied NER method and detail the experimental setup.

4.1 NER method

We use the BERT-based named entity tagger introduced by Luoma and Pyysalo (2020). In brief, the method is based on adding a simple time-distributed dense layer on top of BERT to predict IOB2 named entity tags in a locally greedy manner. The model is both trained and applied with examples consisting of sentences catenated with their context sentences, resulting in multiple predictions for each token (appearing in both “focus” and context sentences). These predictions are then summarized using majority voting. For brevity, we refer to Luoma and Pyysalo (2020) for further details.⁷ Here, we do not use the wrapping of data in documentwise manner as in (Luoma and Pyysalo, 2020), but in bilingual experiments the Finnish and English data are separated with a document boundary token (`-DOCSTART-`) to avoid constructing examples where one input would contain sentences in two languages.

4.2 Experimental setup

The corpora are divided into training, development and test subsets following the subdivisions defined by Pradhan et al. (2013) for OntoNotes, Ruokolainen et al. (2019) for FiNER, and Luoma et al. (2020) for the Turku NER corpus. The newly annotated Finnish law texts are divided chronologically on the document level, placing the earliest-published 48 documents (80%) into training, the latest 6 (10%) into test, and the remaining 6 (10%) into development data. For bilingual experiments, combined training, development and test sets are created by concatenating the corresponding files

⁷The implementation is available from <https://github.com/jouniluoma/bert-ner-cmv>

Type	Train	Dev	Test
ORG	11597	866	2345
PRODUCT	5278	246	1237
DATE	4937	412	889
CARDINAL	4668	439	866
PERSON	4635	488	737
GPE	4127	501	674
ORDINAL	1274	107	190
NORP	1252	115	192
MONEY	909	47	169
LAW	749	154	86
LOC	776	54	120
QUANTITY	611	25	145
PERCENT	642	22	101
TIME	455	35	74
EVENT	326	32	37
WORK_OF_ART	305	56	30
LANGUAGE	219	34	28
FAC	173	20	30

Table 5: Corpus annotation statistics

in each corpus, separating the data for the two languages with a document boundary token.

The hyperparameters are selected based on a grid search following the setup in Luoma and Pyysalo (2020) with the exception that batch size 2 is omitted. The reason for this is that the large combined dataset with a small batch size is too time-consuming on the computational resources available. The parameter selection grid is therefore the following:

- Learning rate: 2e-5, 3e-5, 5e-5
- Batch size: 4, 8, 16
- Epochs: 1, 2, 3, 4

The size of the OntoNotes training set is considerably larger than e.g. that of the previously introduced Finnish corpora, and due to resource limitations (especially GPU computation time), we set the BERT maximum sequence length to 128 WordPiece tokens for all of our experiments.

Parameter selection is performed by evaluating on the development subsets of the corpora. The test sets are held out during preliminary experiments and parameter selection, and are only used to evaluate performance in the final experiments. All of the experiments are repeated 5 times, both for hyperparameter selection and the final test results. The reported results are means and standard deviations calculated from these repetitions. The

Lang.	Prec.	Rec.	F-score
Finnish	92.58 (0.18)	93.41 (0.13)	92.99 (0.14)
English	87.92 (0.20)	89.57 (0.25)	88.74 (0.22)

Table 6: Monolingual NER evaluation results (percentages; standard deviation in parentheses)

hyperparameters for different final models are selected based on their performance on the target language development set as shown in Table 4.

For testing the zero-shot cross-lingual performance on Finnish, we train the mBERT and biBERT models only on the English OntoNotes data and evaluate performance on the Finnish test set. The hyperparameters providing the best results on the English OntoNotes data are used in these experiments, thus reflecting a setting where no annotated Finnish data is available.

5 Results

We next present summary statistics of the newly introduced corpus and then present the results of the machine learning experiments.

5.1 Corpus statistics

Table 5 summarizes the statistics of the new annotation. The combined, extended corpus with the revised OntoNotes-like annotation contains in total nearly 500,000 tokens of text annotated for approximately 55,000 mentions of names and numeric types. While the corpus represents a substantial increase in size and number of annotations over either of the two previously released Finnish NER corpora, the name-annotated subset of the English OntoNotes corpus remains four times larger in terms of token count and over three times larger in terms of the number of annotated entities (Table 1), motivating our exploration of training bilingual models with combined Finnish and English data.

5.2 Monolingual results

Table 6 summarizes the results of monolingual training and evaluation for the FinBERT model on the newly introduced Finnish NER corpus, with results for the original English BERT model on the English OntoNotes results for reference.

For English OntoNotes, the applied method achieves an F-score of 88.74%, comparable to results for similar implementations reported in the literature: for example, Li et al. (2020) re-

Language	Model	Prec.	Rec.	F-score
Finnish	mBERT	89.81 (0.20)	90.76 (0.22)	90.28 (0.17)
Finnish	biBERT	92.47 (0.22)	93.13 (0.11)	92.80 (0.16)
English	mBERT	88.15 (0.20)	89.62 (0.14)	88.88 (0.16)
English	biBERT	88.57 (0.06)	90.03 (0.11)	89.29 (0.07)

Table 7: Bilingual NER model evaluation results (percentages; standard deviation in parentheses)

Type	Monolingual			Bilingual		
	Prec.	Rec.	F-score	Prec.	Rec.	F-score
PERSON	94.12	97.15	95.60	94.92	96.20	95.55
NORP	94.63	96.15	95.36	97.47	96.15	96.80
FAC	67.83	40.00	50.23	70.10	47.33	56.40
ORG	94.14	94.06	94.10	93.97	93.61	93.79
GPE	95.33	97.36	96.33	94.87	97.06	95.95
LOC	87.12	86.50	86.78	86.11	83.67	84.82
PRODUCT	87.53	88.08	87.81	87.11	88.34	87.72
EVENT	72.17	79.46	75.59	69.46	77.84	73.36
WORK_OF_ART	75.00	77.33	75.97	67.52	79.33	72.84
LAW	90.83	96.74	93.69	91.67	94.65	93.13
LANGUAGE	93.05	95.00	94.01	94.95	93.57	94.25
DATE	94.70	94.78	94.74	94.98	95.32	95.15
TIME	81.70	84.32	82.98	78.01	81.35	79.64
PERCENT	95.60	98.61	97.08	100.00	100.00	100.00
MONEY	95.36	94.79	95.08	95.80	91.60	93.65
QUANTITY	87.18	90.90	89.00	86.61	90.07	88.30
ORDINAL	90.33	91.37	90.84	89.56	90.21	89.88
CARDINAL	94.01	95.36	94.68	93.54	95.64	94.58

Table 8: Result details for Finnish data in monolingual setting using FinBERT and bilingual setting using biBERT (percentages)

port 89.16% F-score for *BERT-Tagger* on English OntoNotes 5.0; an approx. 0.4% point difference. While more involved state-of-the-art methods building on BERT have been reported to outperform this result (e.g. 91.11% F-score for the BERT-MRC method of Li et al. (2020)), we are satisfied that the implementation used here is broadly representative of BERT used for NER in a standard sequence tagging setting.

For Finnish, we note that Luoma and Pyysalo (2020) performed an evaluation of the combination of the FiNER and Turku NER corpora with the comparatively coarse-grained six FiNER corpus NE types, reporting an F-score of 93.66% on the combined test set. While not perfectly comparable, the training and evaluation texts of that experiment are strict subsets of the Finnish training and evaluation data here, and we find the F-score of 92.99% on the 18 fine-grained OntoNotes-like annotation a very positive sign of its quality and

consistency: using the newly introduced dataset, we can train models to recognize mentions of *three times as many* name and numeric entity types as previously with only a modest decrease in overall tagging performance.

5.3 Bilingual results

Table 7 summarizes the results of the bi- and multilingual models trained on the combined Finnish and English data and evaluated on the two monolingual corpora. We first observe that the bilingual biBERT model achieves better results than the multilingual mBERT model, providing further support for the findings of Chang et al. (2020) indicating that multilingual training processes produce notably better models when only two languages are targeted. In the remaining, we focus on the results for the biBERT model. For Finnish, we find that the bilingual model fine-tuned on the combined bilingual training data falls just 0.2%

Language	Model	Prec.	Rec.	F-score
Finnish	mBERT	71.00 (0.81)	69.99 (0.47)	70.49 (0.50)
Finnish	biBERT	77.01 (0.47)	77.01 (0.46)	77.01 (0.19)

Table 9: Zero-shot cross-lingual evaluation results from English to Finnish (percentages; standard deviation in parentheses)

points in F-score below the monolingual FinBERT model fine-tuned with monolingual data. For English, we unexpectedly find that the bilingually trained model *outperforms* the monolingual English model with an approx. 0.5% point absolute difference. These results indicate that the annotations of the English OntoNotes NER dataset and the newly introduced Finnish NER dataset are highly compatible, allowing bi- or multilingual methods trained on a bilingual dataset created by their simple concatenation to perform competitively with or even potentially outperform monolingual NER models.

The detailed results presented in Table 8 further show that the performance of the monolingual and bilingual models track very closely, with the monolingual Finnish model slightly outperforming the bilingual for most mention types. An exception to this pattern is seen for NORP, FAC, LANGUAGE, DATE and PERCENT, where the bilingual model shows better performance. These results further suggest that there are no notable annotation inconsistencies in individual types, and that multilingual training may still hold benefit for some entity types.

5.4 Zero-shot cross-lingual results

Finally, Table 9 provides the results of zero-shot cross-lingual transfer from English to Finnish, where a bi- or multilingual model is trained exclusively on English data but then evaluated on Finnish data. We again find that the biBERT model considerably outperforms the mBERT model. While the model performance at 77% falls far behind the over 90% F-scores achieved by the monolingual and bilingual models, it is nevertheless interesting to note that this level of performance can be achieved without any target language data. This cross-lingual transfer approach could potentially be applied e.g. to bootstrap initial annotations for manual revision when creating named entity annotation for languages lacking a corpus annotated with OntoNotes types.

6 Discussion and conclusions

We have introduced a new corpus for Finnish NER created by combining and extending two previously released corpora, FiNER and the Turku NER corpus, and by mapping their custom annotations into the fine-grained OntoNotes representation through a combination of automatic and manual processing steps. The resulting corpus consists of over 50,000 annotations for nearly 500,000 tokens of text representing a broad selection of genres, topics and text types, and is not only the largest resource for Finnish NER created to date, but also identifies three times as many distinct name and numeric entity mention types as the previously introduced Finnish NER corpora.

To assess the internal consistency of the newly created annotation and to provide a baseline for further experiments on the data, we evaluated the performance of a BERT-based NER system initialized with the FinBERT model and fine-tuned on the new Finnish data. These experiments indicated that the annotations of the new corpus can be automatically recognized at nearly 93% F-score, effectively matching previous results with much coarser-grained entity types. To further assess the compatibility of the newly introduced annotation with the original English OntoNotes corpus v5.0 name annotation, we fine-tuned bi- and multi-lingual BERT models on the combination of the Finnish and English corpora, finding that bilingual models can effectively match or potentially even outperform monolingual ones, thus confirming the compatibility of the newly created annotation with existing OntoNotes resources.

All resources introduced in the paper are available under open licenses from <https://github.com/TurkuNLP/turku-one>

Acknowledgments

This work was funded in part by the Academy of Finland. We wish to thank CSC – IT Center for Science, Finland, for computational resources.

References

- Li-Hsin Chang, Sampo Pyysalo, Jenna Kanerva, and Filip Ginter. 2020. Towards fully bilingual deep language modeling. *arXiv preprint arXiv:2010.11639*.
- Rochelle Choenni and Ekaterina Shutova. 2020. What does it mean to be language-agnostic? Probing multilingual sentence encoders for typological properties. *arXiv preprint arXiv:2009.12862*.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota.
- Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. 2014. Building the essential resources for finnish: the turku dependency treebank. *Language Resources and Evaluation*, 48(3):493–531.
- Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60.
- Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.
- Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku neural parser pipeline: An end-to-end system for the conll 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 133–142.
- Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795.
- LDC. 2008. Ace english annotation guidelines for entities. Technical report, Technical report, Linguistic Data Consortium.
- Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859.
- Jouni Luoma, Miika Oinonen, Maria Pyykönen, Veronika Laippala, and Sampo Pyysalo. 2020. A broad-coverage corpus for finnish named entity recognition. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4615–4624.
- Jouni Luoma and Sampo Pyysalo. 2020. Exploring cross-sentence contexts for named entity recognition with BERT. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 904–914.
- Taesun Moon, Parul Awasthy, Jian Ni, and Radu Florian. 2019. Towards lingua franca named entity recognition with bert. *arXiv preprint arXiv:1912.01389*.
- Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 1659–1666.
- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.
- Sampo Pyysalo, Jenna Kanerva, Anna Missilä, Veronika Laippala, and Filip Ginter. 2015. Universal dependencies for finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (Nodalida 2015)*, pages 163–172.
- Vinit Ravishankar, Memduh Gökırmak, Lilja Øvrelid, and Erik Velldal. 2019. Multilingual probing of deep pre-trained contextual encoders. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 37–47, Turku, Finland.
- Teemu Ruokolainen, Pekka Kauppinen, Miikka Silverberg, and Krister Lindén. 2019. A finnish news corpus for named entity recognition. *Language Resources and Evaluation*, pages 1–26.
- Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun’ichi Tsujii. 2012. BRAT: a web-based tool for nlp-assisted

- text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.
- Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.
- Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish. *arXiv preprint arXiv:1912.07076*.
- Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0. *Linguistic Data Consortium, Philadelphia, PA, 23*.
- Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.
- Wei Zhao, Steffen Eger, Johannes Bjerva, and Isabelle Augenstein. 2020. Inducing language-agnostic multilingual representations. *arXiv preprint arXiv:2008.09112*.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

Survey and reproduction of computational approaches to dating of historical texts

Sidsel Boldsen

Dept. of Nordic Studies and Linguistics
University of Copenhagen
Denmark
sbol@hum.ku.dk

Fredrik Wahlberg

Dept. of Linguistics and Philology
Uppsala University
Sweden
fredrik.wahlberg@lingfil.uu.se

Abstract

Finding the year of writing for a historical text is of crucial importance to historical and philological research. However, the year of original creation is rarely explicitly stated and must be inferred from the text content, historical records, and codicological clues. Given a transcribed text, machine learning has successfully been used to estimate years of production. In this paper, we present an overview of estimation approaches from the literature for historical text archives, spanning from the 12th century until today.

1 Introduction

Knowing when a text was written is of crucial importance for relating its content to a historical context. With the increasing digitization of historical archives, many new research opportunities have emerged for studying how languages have evolved. However, such studies rely on digitized corpora explicitly stating when the texts were originally written. This information is often not given by the original scribe, although educated guesses from later owners can sometimes be found in manuscripts. Additionally, improved dating of historical manuscripts can help historians to better understand the chronology of their sources.

The premise for our paper is an imagined scenario where a historian or philologist needs help with a transcribed collection. We imagine being given a partially annotated set of documents (given either as specific years or as intervals) and employing a computer model to determine the production years of the un-

labelled documents. Although there is literature describing different ways of solving the problem of the above scenario, there is little work done on comparing the different modelling approaches. In this paper, we will survey and evaluate computational approaches to the problem of estimating the production dates of text in digitalized historical archives. We have reimplemented several methods for estimation and feature extraction proposed in the literature. Our experimental setup allows us to evaluate combinations of different methods on datasets representing different times, text lengths, and genres. Our reimplementations are available as open source¹.

Our primary historical datasets were two medieval archives containing legal documents from Denmark and Sweden. Comparing results on these collections is of special interest, as they are similar with respect to content, but differ in the number of documents, temporal distributions and detail of annotation. To assess the generalizability of the methods we also include two modern collections. These modern collections, that have previously been treated in the literature, are a collection of English news items, from the SemEval 2015 shared task on diachronic text evaluation (Popescu and Strapparava, 2015), and Colonia, a corpus of historical Portuguese (Zampieri and Becker, 2013).

An overview of the relevant literature is presented in Section 2, Section 3 contains a description of the datasets, our experimental setup is described in Section 4, and, finally, results and discussion are presented in Section 5.

¹Python notebooks can be found at <http://github.com/fredrikwahlberg/nodalida21>

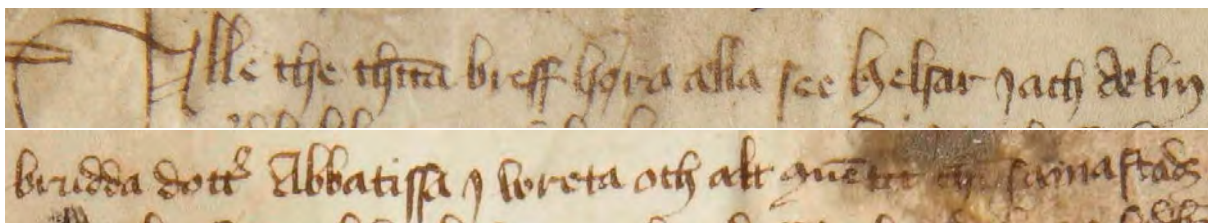


Figure 1: The first line of charter SDHK 18863, containing an agreement on an exchange of land in 1417. The text is "Alle the thetta breff hœra ælla see helsar jach Ælin Bruddadotter, abbatissa i Wreta, och alt conuentit ther samastadz" (from "Svenskt Diplomatariums Huvudkartotek" 18863, section 3).

2 Previous work

The problem of automatic text dating has been treated using various methods and applied to a wide range of different types of corpora. To the best of our knowledge, the task of assigning a date to documents was first introduced in the information retrieval community with the main goal to query document collections based on temporal relevance. De Jong et al. (2005) treat the problem as a text classification task in which documents are dated by comparing them to temporal query profiles. They refer to such profiles as *temporal language models*, which essentially capture the distribution of term or concept usage over time. The same idea is found in the work of Dalli and Wilks (2006), in which word frequencies across time are used to infer temporal association rules. The work by de Jong et al. (2005) was later expanded by Kanhabua and Nørvåg (2008) who improved the temporal language models by applying various steps of pre-processing, including filtering words based on TF-IDF scores and POS tags, applying stemming, and collocation extraction.

The above works were made on corpora of newspaper articles, which have remained to be an object of research within the field of automatic dating, most recently in the SemEval 2015 shared task on diachronic text evaluation (DTE) on English news snippets (Popescu and Strapparava, 2015). Following the work on temporal language modelling, Garcia-Fernandez et al. (2011) introduced using support vector machines (SVM) for the task of dating, in which documents are represented by feature vectors of word and character counts, in addition to other handcrafted features. Whereas in the temporal language

modelling approach the sole problem is to learn the distribution of words in a set of documents belonging to a specific time span, the goal of mapping a document to a date is now part of the learning objective. Later work on news corpora has focused on how the extraction of temporal references, such as expressions for time and events, can facilitate the task of dating, which was also the research question in two of the three subtasks of the DTE shared task (Chambers, 2012; Vashishth et al., 2019).

Aside from news, scholars have studied a wide range of different historical corpora, ranging from broad collections such as Google n-grams (Popescu and Strapparava, 2014) to more narrow collections as in the DaDoEval2020 shared task (Menini et al., 2020), which introduced a diachronic corpus of political work by Alcide De Gasperi. While news items naturally contain explicit temporal references for when the text was written, this is often not the case when working with other genres. For example, if a philologist were to date a piece of literature, their work may solely rely on features such as lexicon, grammar, topic, or style, as the contemporary context is often implicit. Thus, work outside the news genre has generally put less emphasis on extracting temporal references, and instead explore how the language in a text can be represented.

One of the first studies to extend the work beyond the news genre was Kumar et al. (2011), who use language modelling to predict the date of a collection of short stories published between 1798 to 2008 from Project Gutenberg². Subsequently, language modelling has not been applied to the problem of

²<https://www.gutenberg.org>

dating. Work has been done to identify temporal trends in historical corpora (Pichel Campos et al., 2018; Pichel et al., 2020; Boldsen et al., 2019), by using language modelling to measure the distance between time periods, but models were not explicitly applied to the task of dating. Instead, studies have focused on creating vector representations of documents and then using those for classification. The raw text has been used directly as input to create bag-of-words and/or characters, with n-gram sizes ranging from one to three words (Niculae et al., 2014; Szymanski and Lynch, 2015; Zampieri et al., 2016), and one to five characters (Garcia-Fernandez et al., 2011; Niculae et al., 2014; Szymanski and Lynch, 2015). Other features may be extracted, such as syntactic features using POS annotations (Szymanski and Lynch, 2015; Zampieri et al., 2015, 2016) and stylistic measures such as lexical diversity (Štajner and Zampieri, 2013; Zampieri et al., 2015).

Most commonly, the problem of dating a text is defined as a classification problem in which classes are treated as bins corresponding to different time spans. Several estimators have been applied, including logistic regression (Chambers, 2012), support vector machines (Garcia-Fernandez et al., 2011; Szymanski and Lynch, 2015; Zampieri et al., 2016) and multinomial naive Bayes (Mihalcea and Nastase, 2012; Zampieri et al., 2016). The size of the bins depend on the problem and the data available. For dating of contemporary news items, scholars have worked with granularities down to a yearly basis (Chambers, 2012; Vashishth et al., 2019). This is typically not possible when working with historical text, as data is sparse and may in turn not have such a precise date of production. Here, scholars have instead worked on dating documents within a century (Štajner and Zampieri, 2013) or a decade (Popescu and Strapparava, 2015).

Compared to classification, regression methods have not been extensively explored. In regression, samples are mapped to a date directly instead of a bin, thus circumventing the obstacle of deciding on a specific bin size. Also, regression preserves the ordinal nature of the problem, which classification ignores. Niculae et al. (2014) propose to use ordinal regression.

In this approach, the task of dating is considered as a ranking problem, where reference documents are placed on a timeline, which is then used to estimate the most probable time spans for query documents. Another attempt using regression comes from the field of image processing, where Wahlberg et al. (2016) applied Gaussian Processes (GP) to the problem of estimating the date of medieval manuscripts using visual features extracted from the facsimile together with the transcribed text.

Whether classification or regression is best suited for the problem of dating text - and what pitfalls such approaches have - are still open questions. As for feature extraction, neural methods have over the last decades undermined the use of manual feature extraction for a wide range of problems, including text classification. Vashishth et al. (2019) applied graph convolutional networks to the problem of dating, utilizing syntactic information and temporal reference extraction in addition to the words of the text. For smaller corpora, neural approaches are yet to be tested, which is out of the scope of this paper. Instead, we seek to describe and compare the methods that have already been established for the dating of historical text corpora.

3 Datasets

3.1 Svenskt Diplomatariums Huvudkartotek

”Svenskt Diplomatariums Huvudkartotek” (SDHK) is a collection of charters from medieval Sweden (c. 1050-1523). The collection consists of approximately 44,000 charters on (mostly) parchment, of which about 10,500 have been transcribed. The most frequent languages are Swedish (c. 3,000 transcribed charters) and Latin (c. 7,500 transcribed charters). Of the full collection, about 11,000 charters have been photographed, largely overlapping with the transcribed set.

While the Latin vocabulary and spelling are fairly consistent, except for small variations in the use of abbreviations, the Swedish text changes significantly with time. The Swedish language goes through significant development from Old Swedish (”fornsvenska”) involving grammar, lexicon, and spelling between the 13th and 16th centuries. The material is fur-

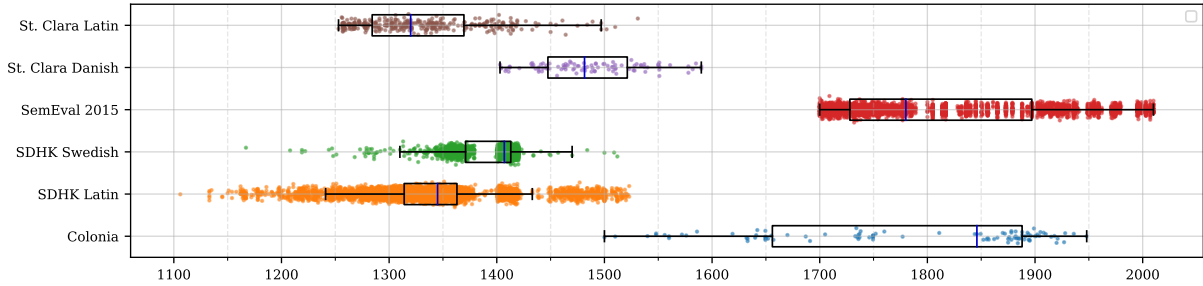


Figure 2: Box plots showing the represented years for the document collections (coloured dots are individual documents) in each dataset described in section 3.

ther complicated by the transcribers’ inconsistent expansion of abbreviations and spelling normalization. Since these types of problems are common with this type of archive, we did not see any need for further annotation or human curation. Hypothetically, if many researchers, each primarily interested in their limited period, have contributed to the transcribed collection, then transcription standards (e.g., expansions of abbreviations) might change over time, through not due to changes in the underlying historical material. Though a machine could potentially overfit on such features, we see this as falling outside the scope of this paper.

3.2 The charters of St. Clara Convent

The charters of St. Clara Convent (Roskilde, Denmark) are part of the Arnamagnæan Collection at the University of Copenhagen (Hansen, 2015). The charters date from when the convent was founded in 1256 till it was closed after the Reformation, after which the convent and its archive became part of the university’s properties. In total, 471 charters are left from the old archive. The majority of the charters are written in Latin and Danish (361 and 100 charters, respectively), the rest being in German or Swedish.

The charters have been all digitized with multiple layers of annotation, including both a *facsimile* and *diplomatic* transcription of the text. The facsimile level (a) captures the handwritten form of the text by annotating the palaeographic characteristics of the letters (i.e., focusing on the shape of the character rather than solely on its meaning). The diplomatic transcription (b) is of the kind that is usually found in manuscript editions. At this

level, the difference in handwriting is ignored and abbreviated diacritics are expanded, while variation in spelling is still preserved:

- (a) fo2o2 ʒ monafterij earū in posterum
- (b) soror(um) (et) monasterij earu(m) in posterum

In the example above, the word ”fo2o2” is written in a way very similar to the original handwriting (i.e., as a facsimile). In diplomatic annotation, this becomes ”soror(um)”, where ”soror” are the modern forms of the letters and the ”-um” suffix is expanded from the stroke on the last letter and inferred from the context.

3.3 SemEval2015

The SemEval2015 shared task of ”Diachronic Text Evaluation” introduces a corpus of English news snippets dating from the 18th to the 21th centuries (Popescu and Strapparava, 2015). Contrary to the collections of charters, the news snippets were not precisely dated but rather given as intervals over years (2 and 6 years wide) which can be seen by the distribution of data points in Figure 2. For this paper, we only utilize the training set data from the task, ending up with c. 4,500 documents.

3.4 Colonia

Colonia is a corpus of historical Portuguese, compiled from various sources spanning from the 16th to the 20th century (Zampieri and Becker, 2013). While the collection of news snippets and charters contains text with lengths ranging from 10 to hundreds of words, the texts of Colonia are substantially longer, containing full works with thousands of tokens.

Thus, with the 100 documents that it contains, the collection counts up to five million tokens in total.

4 Experimental setup

In our experimental setup, we have implemented a number of ways of doing feature extraction. We then evaluated all combinations between those feature spaces and a number of ways of doing the mapping to years on the timeline.

In the documents of several datasets, clearly stated years can be found. In order not to let the estimators simply learn to find this information (especially in the charter datasets, where Roman numerals are frequently encountered) we have removed all numerals from the text as a part of the preprocessing.

Some of the methods we have evaluated were quite demanding of the hardware. Hence, we randomised the training, validation, and test sets while preprocessing the datasets, using the same sets for all evaluations. It should be noted that this is not standard for several of our approaches (e.g., naive Bayes) which are normally evaluated using cross-validation. However, we saw this as the only way of making a fair comparison and not risk giving some estimators more or different data.

4.1 Feature spaces

Binary bag-of-words vectors (BOW) (i.e., encoding the existence or absence of a word) have been shown to be useful in many applications. Since the popularity of words changes over time, this type of vector can encode distributional information on word choice. We generated such vectors from the training and validation folds of the datasets and then transformed the full datasets into their respective vector space representations. This meant that only the part of the test set vocabulary that was overlapping with the training and validation vocabularies was used. As the Colonia dataset had a higher level of annotation, we made binary BOW vectors from the words, pos-tags, and concatenated word+pos-tags.

Several papers use n-gram feature vectors on both the word and character level. Looking at a small context around words has the potential to encode changes in common ex-

pressions or even some semantics. In contrast, character level n-grams have the potential to catch spelling or phonetic changes (especially during eras where there were no standardised spellings). The order of a space spanned by n-grams is only limited by computer memory. We chose to extract n-grams of orders $\{1, 2, 3\}$.

For some estimators, it is considered best practice to perform feature selection as noise removal and to lower run times. We ran feature selection based on χ^2 statistics, capping the feature space dimensionality to 1000 for all estimators, but only kept the automatic selection for those estimators where the training accuracy improved (Gaussian process, linear SVM, and non-linear SVM).

4.2 Classification for date estimation

Usually, the date estimation was treated as a classification task in the literature. This was done by formulating the mapping from documents to the timeline by dividing the timeline 25-year wide bins and then classifying the documents into those bins. An advantage of this approach was that several estimators can be used, specializing on particular parts of the timeline (Garcia-Fernandez et al., 2011; Zampieri et al., 2016).

The most popular estimation method in our chosen literature is the support vector machine (SVM) (Cortes and Vapnik, 1995). One core advantage with the SVM is that finding a separation in some feature space is a reasonable fast convex optimisation problem. The resulting linear decision boundary is interpretable in term of the feature set, especially with BOW vectors, but suffers from the fact that the data needs to be linearly separable. In the literature, strategies for finding hyper-parameters or kernels are surprisingly absent. From this, we draw the conclusion that (most likely) a linear SVM was used, which only has one regularisation hyper-parameter. Because of the high dimensionality of some feature spaces, a non-linear decision boundary is often not needed (and expensive). For testing this in our setting, we extended our experiments by using the standard radial basis function (RBF) kernel to introduce some non-linearity, in addition to the linear SVM.

Temporal language models are probabilistic models over sequences of tokens, either words

or characters, for a given set of time spans. The model approximates the likelihood of a sequence, given some corpus. To simplify such models, the Markov assumption is commonly used to split up longer sequences, creating a so-called n-gram model (as in the feature described above). In order to create temporal language models for classification, we split up the data into bins and trained language models on these respective bins (Boldsen and Paggio, 2019). Given this set of temporal language models, dating a document is equivalent to finding the model that is more likely to generate a specific document. One of the issues in estimating sequence probabilities is encountering unseen n-grams. This is commonly handled by modifying the n-gram counts by discounting from non-zero events. In this paper, we used modified Kneser-Ney smoothing with interpolation (Chen and Goodman, 1999).

Naive Bayes classifiers (surprisingly) often deliver good results in a variety of domains despite their assumption of independence between features. Zampieri et al. (2016) employ a multinomial naive Bayes classifier, which is common for linguistic applications. This fits well with their chosen feature model, focusing on the frequencies of words and POS-tags. For the completeness of the comparison, we evaluate estimators using both multinomial and Gaussian priors.

4.3 Regression for date estimation

To get around the problem of choosing the proper bin width for a classification, some papers treat dating as a regression problem. In Wahlberg et al. (2016), a Gaussian process (GP) was used for the regression, allowing mapping from documents to normal distributions over the timeline (i.e., inferring uncertainties in addition to point estimates).

For a GP, the weight vector ω , in the standard regression expression $\hat{y}_i = \omega\phi(x_i)$, is treated as a random vector from a multivariate normal distribution (Rasmussen and Williams, 2006). Though the GP is non-parametric and ω is analytically inferred from the data, the hyper-parameters for the feature transform (kernel) $\phi(\cdot)$ must be trained (we used RBF as to be able to compare to the SVM) by maximizing the likelihood of generating the training data given that parameter set. Since

	MVB Uniform Weighted		
Colonia	26.32	5.65	11.74
SDHK Latin	26.29	5.89	17.69
SDHK Swedish	69.04	7.16	54.24
SemEval 2015	23.64	7.72	11.32
St.Clara dipl. Danish	15.79	12.48	14.27
St.Clara dipl. Latin	19.72	8.31	14.71
St.Clara facs. Danish	15.79	12.49	13.85
St.Clara facs. Latin	19.72	8.25	14.68

Table 1: Accuracy for different baseline strategies. The majority vote baseline (MVB) classifies all documents as the most common class while the other baselines are expected accuracy with hypothetical random classifier. The "uniform" baseline classifier draws random years from a uniform distribution over the relevant timeline, while the "weighted" draws from each dataset's label distribution.

GPs are generative and probabilistic, all hyperparameters can be marginalized. However, this is rarely done in practice. Most often, a set of hyperparameters are chosen by maximizing their likelihood given the model (maximum a posteriori).

4.4 Evaluation metric

In most of the papers presented in Section 2, accuracy was the preferred evaluation metric. For any classification over a timeline, a bin width needs to be chosen. Several papers used 50-year-wide non-overlapping bins. In our implementation we have chosen 25-year wide bins, making accuracy less forgiving.

As for accuracy baselines, we created random baseline classifications using three strategies. First, the majority vote baseline (i.e., always classifying as the most common bin), a uniform bin probability, and a weighted scheme with random classifications while respecting the date distribution of the data. The baseline accuracy scores can be found in Table 1. Given these methods, any accuracy above 25% can be seen as better than random for all datasets except for SDHK in Swedish, which is heavily skewed and has a majority vote baseline of 69%.

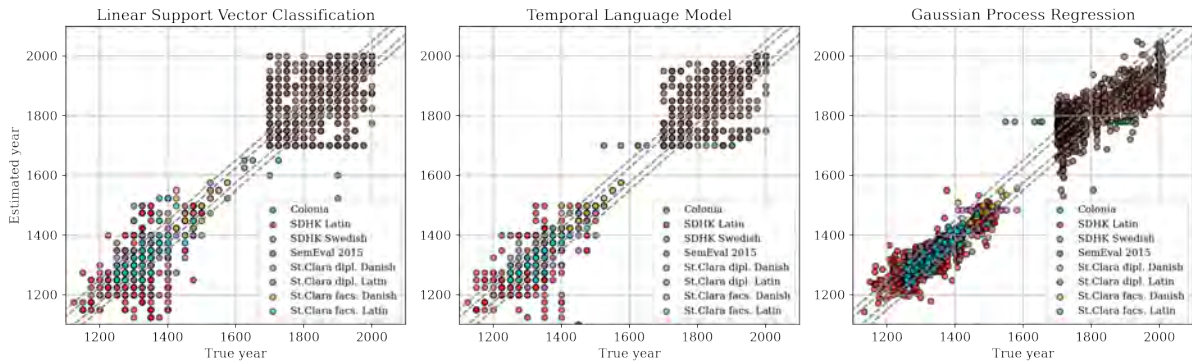


Figure 3: Scatter plot over the estimated production years versus their true years for three types of estimators. All used character bigram features and classification except for the rightmost that used regression. The dashed lines on the diagonal are spaced 25 and 50 years from the diagonal. Note that many points are plotted on top of each other, especially for the classification based estimators.

5 Results and discussion

The results from the experiments can be found in Table 2. Due to a lack of space, we chose to focus the discussion on the four different classifiers that provide the highest scores for the individual datasets (highlighted with red). The results for the remaining two classifiers can be found in the appendix.

All classifiers perform above baseline for at least one feature set. Considering the best performing feature sets per classifier (highlighted with blue), character models perform the best across classifiers except for Gaussian Naive Bayes. Aside from being able to capture features such as morphology and spelling, character models have the advantage that the feature space is smaller than for word models, which in turn increases the number of examples that estimators consider. Whether it is the features or simply the data size that is at play is difficult to read from these numbers.

When working with vector representation of words and higher level character n-grams, the feature set easily becomes larger than the number of samples used for training a model. In these cases, one could argue that it is unlikely for the estimators using these representations (SVMs, naive Bayes) *not* to find something in the training set that correlates with the timeline, even though the feature might not necessarily be related to language change. The problem is compounded by that the training, validation, and test data were all drawn from

the same data generating process and, hence, might have the same spurious correlations in relation to the target labels.

If we compare the linear SVM with the non-linear SVM, the linear version has the advantage of being more qualitatively interpretable due to the lack of warping of the feature space. However, if we compare the models in terms of accuracy, using a non-linear kernel yields slightly better results. When we compare the test set predictions of the different estimators, they do tend to correlate. As is revealed in Figure 4, there is a strong relationship between the predictions made using different SVM estimators (linear and non-linear), especially on similar feature sets. If we consider the predictions using the non-linear SVM on character unigrams, we see a slightly stronger correlation with the predictions of the linear SVM when using higher orders, which suggests that a more complex model is able to utilize its non-linear combination of features on the problem. However, in terms of accuracy results, this advantage is not widely outspoken. Thus, we argue that choosing a linear kernel may still be preferable, as its predictions are more easily explained to a community of philologists or historians.

Despite not appearing in more recent research, the temporal language model outperforms other models on several datasets using character features. All estimators that we have evaluated describe language as a distribution of words or characters. What distin-

Temporal Language Model Classification						
	char ₁	char ₂	char ₃	word ₁	word ₂	word ₃
Colonia	5.3	5.3	5.3	5.3	5.3	5.3
SDHK Latin	0.5	69.2	75.3	0.0	7.5	15.7
SDHK Swedish	1.9	93.5	95.0	0.3	1.0	5.7
SemEval 2015	25.7	45.7	58.3	1.2	15.6	16.5
St. Clara dipl. Danish	15.8	42.1	31.6	0.0	31.6	31.6
St. Clara dipl. Latin	1.4	54.9	56.3	0.0	26.8	29.6
St. Clara facs. Danish	42.1	63.2	57.9	5.3	10.5	10.5
St. Clara facs. Latin	7.0	69.0	71.8	0.0	1.4	2.8
Linear Support Vector Classification						
Colonia	36.8	47.4	36.8	36.8	42.1	36.8
SDHK Latin	37.5	53.9	53.4	41.3	35.1	34.2
SDHK Swedish	81.0	89.0	88.0	76.2	69.9	69.4
SemEval 2015	26.8	31.4	30.2	28.4	24.7	24.2
St. Clara dipl. Danish	26.3	57.9	36.8	10.5	10.5	10.5
St. Clara dipl. Latin	38.0	47.9	39.4	32.4	19.7	26.8
St. Clara facs. Danish	42.1	31.6	10.5	0.0	10.5	21.1
St. Clara facs. Latin	47.9	49.3	36.6	33.8	15.5	22.5
Gaussian naive Bayes						
Colonia	31.6	21.1	31.6	26.3	36.8	42.1
SDHK Latin	12.9	37.2	58.3	62.9	-	-
SDHK Swedish	19.8	84.6	92.7	86.9	88.8	-
SemEval 2015	19.7	21.7	39.8	50.9	49.0	43.1
St. Clara dipl. Danish	31.6	26.3	21.1	36.8	47.4	15.8
St. Clara dipl. Latin	16.9	39.4	54.9	54.9	66.2	69.0
St. Clara facs. Danish	26.3	31.6	36.8	36.8	47.4	36.8
St. Clara facs. Latin	53.5	67.6	70.4	63.4	66.2	59.2
Support Vector Classification with Radial Basis Function						
Colonia	42.1	52.6	42.1	42.1	42.1	42.1
SDHK Latin	45.0	53.4	58.3	48.0	40.1	10.6
SDHK Swedish	88.3	90.3	90.1	80.6	1.3	1.5
SemEval 2015	27.6	30.1	31.6	27.4	10.3	14.3
St. Clara dipl. Danish	26.3	57.9	26.3	21.1	15.8	21.1
St. Clara dipl. Latin	45.1	46.5	50.7	38.0	25.4	19.7
St. Clara facs. Danish	36.8	21.1	31.6	21.1	15.8	26.3
St. Clara facs. Latin	50.7	47.9	45.1	33.8	15.5	25.4

Table 2: The accuracy scores (in percent) for the four estimators. Best results for each dataset are highlighted with red, and best results for each estimator are highlighted with blue. We ran several more combinations of feature sets and estimators, all of which can be found in our code repository for this paper.

guishes the temporal language modelling approach from the other estimators, is that it uses perplexity as a measure to model linguistic difference. Several estimators are treating the probability density functions for the different documents as points in a Euclidean space (e.g., linear SVM). This assumption often works. However, by using a divergence metric between probability density functions, the space is treated more in line with the nature of the encoding. This has been shown to be beneficial for image based dating of manuscripts (Wahlberg et al., 2014), leading us to speculate that this result is valid here too.

While performing well on character feature sets, the temporal language model struggles when it comes to word representations with accuracies below 10%. This suggests that the temporal language model is sensitive to larger feature spaces, in which smoothing might not be sufficient. Furthermore, it performs poorly on the Colonia dataset. Whether this is due to the number of samples, document length, or dataset distribution is difficult to say, and it calls for further analysis of the models with respect to dataset statistics.

Finally, we wish to discuss the performance of regression to classification methods. Most previous work has preferred to use classifica-

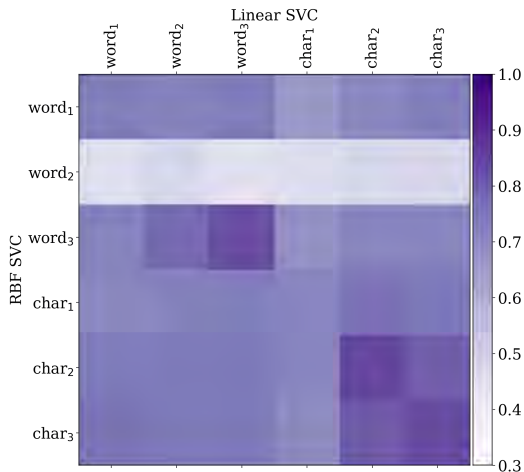


Figure 4: A heat map of the correlation coefficients ($p < 0.005$) between test set predictions by SVM estimators with linear and RBF kernels using different feature sets. The coefficients ($p < 0.005$) were computed as Kendall’s τ , which does not assume a normal distribution and works for ordinal values.

tion instead of regression, treating the timeline as discrete and with temporally independent labels. That labels are independent is reflected in the use of categorical accuracy as the evaluation metric. If we look at Figure 3, this is illustrated by the inner dashed lines, outside which predictions are considered incorrect, even though they are close to the target temporally. In this respect, regression methods should have an advantage, however, this is not reflected in our results. It would be interesting to further compare what advantages there are - if any - to choosing regression over the classification.

6 Conclusion and Future Work

In this paper, we present a survey of several methods found in the literature for estimating the production years of transcribed historical documents. We have reproduced the methods used in a number of papers, including different n-gram/word/pos-tag feature spaces and several linear (naive Bayes, linear SVM) and non-linear (Gaussian process, SVM with RBF kernel) estimators.

Our results show that several of the combinations of estimators and feature models work well, but that character n-gram features provide the best results overall. In particular, the

temporal language model with character features surpasses more recently proposed models. Whether this is due to the linguistic features (e.g., suffixes or phonetic changes leading to changes in spelling) that they potentially capture or simply due to a reduced feature space giving better model parameter estimates, we cannot conclude from our results. Therefore, we call for further analysis of the estimators, preferably favouring more interpretable approaches (e.g., linear SVM).

Our experiments show that combinations of estimators and feature transforms that worked well on younger materials were often also successful on older materials, and vice versa. As the datasets that we compare not only differ in age, but also in number and size of samples. For future work, it would be interesting to investigate the robustness of the methods from the literature with respect to such dataset statistics. In this respect, it would also be relevant to include recent work on neural models such as using word embeddings and convolutional networks, which have been shown to work well for dating on large corpora. However, these have yet to be trialed on smaller corpora.

Acknowledgments

The first author is supported by the project *Script and Text in Time and Space*, a core group project supported by the Velux Foundations. We are grateful to Patrizia Paggio for her support and comments regarding this paper.

We also want to thank the Swedish National Archive for providing the SDHK dataset, both as images and transcribed text. Funding for the second author was provided by the project “New Eyes on Sweden’s Medieval Scribes”, headed by Lasse Mårtensson.

Finally, we want to thank the anonymous reviewers for finding the time to give constructive criticism.

References

- Sidsel Boldsen, Manex Agirrezabal, and Patrizia Paggio. 2019. Identifying temporal trends based on perplexity and clustering: Are we looking at language change?
- Sidsel Boldsen and Patrizia Paggio. 2019. Automatic dating of medieval charters from denmark. In *DHN*.
- Nathanael Chambers. 2012. Labeling documents with timesteps: Learning from their time expressions. Technical report, NAVAL ACADEMY ANNAPOLIS MD DEPT OF COMPUTER SCIENCE.
- Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394.
- Corinna Cortes and Vladimir Vapnik. 1995. <https://doi.org/10.1023/A:1022627411411> Support-vector networks. *Machine Learning*, 20(3):273–297.
- Angelo Dalli and Yorick Wilks. 2006. Automatic dating of documents and temporal text classification. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 17–22.
- Anne Garcia-Fernandez, Anne-Laure Ligozat, Marco Dinarelli, and Delphine Bernhard. 2011. When was it written? Automatically determining publication dates. In *String Processing and Information Retrieval*, pages 221–236, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Anne Mette Hansen. 2015. Adkomstbreve i Skt. Clara Klosters arkiv. In Matthew J. Driscoll and Svanhildur Óskarsdóttir, editors, *66 håndskrifter fra Arne Magnussons samling*, pages 138–139. Museum Tusulanum.
- Franciska de Jong, Henning Rode, and Djoerd Hiemstra. 2005. Temporal language models for the disclosure of historical text. In *Humanities, computers and cultural heritage: Proceedings of the XVIth International Conference of the Association for History and Computing (AHC 2005)*, pages 161–168. Koninklijke Nederlandse Academie van Wetenschappen.
- Nattiya Kanhabua and Kjetil Nørvåg. 2008. Improving temporal language models for determining time of non-timestamped documents. In *International Conference on Theory and Practice of Digital Libraries*, pages 358–370. Springer.
- Abhimanu Kumar, Matthew Lease, and Jason Baldridge. 2011. Supervised language modeling for temporal resolution of texts. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2069–2072.
- Stefano Menini, Giovanni Moretti, and S. Tonelli R. Sprugnoli. 2020. Dating document evaluation at EVALITA 2020. <https://dhfbk.github.io/DaDoEval/>. Accessed: 2020-08-03.
- Rada Mihalcea and Vivi Nastase. 2012. Word epoch disambiguation: Finding how words change over time. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 259–263.
- Vlad Niculae, Marcos Zampieri, Liviu P Dinu, and Alina Maria Ciobanu. 2014. Temporal text ranking and automatic dating of texts. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 17–21.
- José Ramom Pichel, Pablo Gamallo, Iñaki Alegria, and Marco Neves. 2020. <https://doi.org/10.1080/09296174.2020.1732177> A methodology to measure the diachronic language distance between three languages based on perplexity. *Journal of Quantitative Linguistics*, 0(0):1–31.
- José Ramom Pichel Campos, Pablo Gamallo, and Iñaki Alegria. 2018. <http://aclweb.org/anthology/W18-3916> Measuring language distance among historical varieties using perplexity. Application to European Portuguese. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 145–155. Association for Computational Linguistics.
- Octavian Popescu and Carlo Strapparava. 2014. Time corpora: Epochs, opinions and changes. *Knowledge-Based Systems*, 69:3–13.
- Octavian Popescu and Carlo Strapparava. 2015. Semeval 2015, task 7: Diachronic text evaluation. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 870–878.
- C. E. Rasmussen and C. K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. MIT Press.
- Sanja Štajner and Marcos Zampieri. 2013. Stylistic changes for temporal text classification. In *Text, Speech, and Dialogue*, pages 519–526, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Terrence Szymanski and Gerard Lynch. 2015. UCD: Diachronic text classification with character, word, and syntactic n-grams. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 879–883.

- Shikhar Vashishth, Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2019. <http://arxiv.org/abs/1902.00175> Dating documents using graph convolution networks.
- F. Wahlberg, L. Mårtensson, and A. Brun. 2014. <https://doi.org/10.1109/ICFHR.2014.128> Scribal attribution using a novel 3-d quill-curvature feature histogram. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 732–737.
- F. Wahlberg, L. Mårtensson, and A. Brun. 2016. Large scale continuous dating of medieval scribes using a combined image and language model. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 48–53.
- Marcos Zampieri and Martin Becker. 2013. Colonia: Corpus of historical portuguese. *ZSM Studien, Special Volume on Non-Standard Data Sources in Corpus-Based Research*, 5:69–76.
- Marcos Zampieri, Alina Maria Ciobanu, Vlad Niculae, and Liviu P Dinu. 2015. Ambra: A ranking approach to temporal text classification. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 851–855.
- Marcos Zampieri, Shervin Malmasi, and Mark Dras. 2016. Modeling language change in historical corpora: The case of Portuguese. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4098–4104, Paris, France. European Language Resources Association (ELRA).

A Experimental results

	Temporal Language Model Classification					
	char ₁	char ₂	char ₃	word ₁	word ₂	word ₃
Colonia	5.3	5.3	5.3	5.3	5.3	5.3
SDHK Latin	0.5	69.2	75.3	0.0	7.5	15.7
SDHK Swedish	1.9	93.5	95.0	0.3	1.0	5.7
SemEval 2015	25.7	45.7	58.3	1.2	15.6	16.5
St. Clara dipl. Danish	15.8	42.1	31.6	0.0	31.6	31.6
St. Clara dipl. Latin	1.4	54.9	56.3	0.0	26.8	29.6
St. Clara facs. Danish	42.1	63.2	57.9	5.3	10.5	10.5
St. Clara facs. Latin	7.0	69.0	71.8	0.0	1.4	2.8
Linear Support Vector Classification						
Colonia	36.8	47.4	36.8	36.8	42.1	36.8
SDHK Latin	37.5	53.9	53.4	41.3	35.1	34.2
SDHK Swedish	81.0	89.0	88.0	76.2	69.9	69.4
SemEval 2015	26.8	31.4	30.2	28.4	24.7	24.2
St. Clara dipl. Danish	26.3	57.9	36.8	10.5	10.5	10.5
St. Clara dipl. Latin	38.0	47.9	39.4	32.4	19.7	26.8
St. Clara facs. Danish	42.1	31.6	10.5	0.0	10.5	21.1
St. Clara facs. Latin	47.9	49.3	36.6	33.8	15.5	22.5
Gaussian naive Bayes						
Colonia	31.6	21.1	31.6	26.3	36.8	42.1
SDHK Latin	12.9	37.2	58.3	62.9	-	-
SDHK Swedish	19.8	84.6	92.7	86.9	88.8	-
SemEval 2015	19.7	21.7	39.8	50.9	49.0	43.1
St. Clara dipl. Danish	31.6	26.3	21.1	36.8	47.4	15.8
St. Clara dipl. Latin	16.9	39.4	54.9	54.9	66.2	69.0
St. Clara facs. Danish	26.3	31.6	36.8	36.8	47.4	36.8
St. Clara facs. Latin	53.5	67.6	70.4	63.4	66.2	59.2
Support Vector Classification with Radial Basis Function						
Colonia	42.1	52.6	42.1	42.1	42.1	42.1
SDHK Latin	45.0	53.4	58.3	48.0	40.1	10.6
SDHK Swedish	88.3	90.3	90.1	80.6	1.3	1.5
SemEval 2015	27.6	30.1	31.6	27.4	10.3	14.3
St. Clara dipl. Danish	26.3	57.9	26.3	21.1	15.8	21.1
St. Clara dipl. Latin	45.1	46.5	50.7	38.0	25.4	19.7
St. Clara facs. Danish	36.8	21.1	31.6	21.1	15.8	26.3
St. Clara facs. Latin	50.7	47.9	45.1	33.8	15.5	25.4
Multinomial Naive Bayes						
Colonia	26.3	26.3	26.3	26.3	26.3	26.3
SDHK Latin	26.3	26.4	29.9	39.0	39.1	36.7
SDHK Swedish	69.0	69.0	69.0	69.0	69.0	69.0
SemEval 2015	23.6	23.6	23.6	23.6	23.6	23.6
St. Clara dipl. Danish	10.5	21.1	15.8	26.3	21.1	10.5
St. Clara dipl. Latin	19.7	19.7	19.7	22.5	21.1	19.7
St. Clara facs. Danish	26.3	26.3	26.3	26.3	21.1	26.3
St. Clara facs. Latin	25.4	25.4	19.7	25.4	19.7	19.7
Gaussian Process Regression						
Colonia	21.1	0.0	31.6	0.0	0.0	5.3
SDHK Latin	34.1	35.6	38.9	40.9	28.8	26.6
SDHK Swedish	79.7	75.4	79.3	80.2	3.1	1.5
SemEval 2015	12.8	17.1	15.3	12.3	8.3	6.9
St. Clara dipl. Danish	15.8	21.1	47.4	31.6	26.3	21.1
St. Clara dipl. Latin	23.9	22.5	29.6	25.4	16.9	16.9
St. Clara facs. Danish	21.1	47.4	52.6	21.1	26.3	21.1
St. Clara facs. Latin	54.9	42.3	42.3	31.0	14.1	16.9

Multilingual and Zero-Shot is Closing in on Monolingual Web Register Classification

Samuel Rönqvist* Valtteri Skantsi*[◦] Miika Oinonen* Veronika Laippala*

*TurkuNLP, University of Turku, Finland

[◦]NSE, University of Oulu, Finland

{saanro, valtteri.skantsi, mhtoin, mavela}@utu.fi

Abstract

In this paper, we present experiments in register classification of documents from the unrestricted web, such as news articles or opinion blogs, in a multilingual setting, exploring both the benefit of training on multiple languages and the capabilities for zero-shot cross-lingual transfer. While the wide range of linguistic variation found on the web poses challenges for register classification, recent studies have shown that good levels of cross-lingual transfer from the extensive English CORE corpus to other languages can be achieved. In this study, we show that training on multiple languages 1) benefits languages with limited amounts of register-annotated data, 2) on average achieves performance on par with monolingual models, and 3) greatly improves upon previous zero-shot results in Finnish, French and Swedish. The best results are achieved with the multilingual XLM-R model. As data, we use the CORE corpus series featuring register annotated data from the unrestricted web.

1 Introduction

The focus of this paper is on multilingual training and cross-lingual transfer in register classification of web documents. Text register (or genre) (Biber, 1988), such as discussion forum or encyclopedia article, has been shown to be one of the most important predictors of linguistic variation (Biber, 2012), and register affects also the automatic processing of text (Mahajan et al., 2015; Webber, 2009; Van der Wees et al., 2018). Yet, web data is typically used without register information in many NLP tasks.

Web register classification studies have suffered from the lack of corpora featuring the full range of

registers found on the web, as many datasets are based on a priori selection of register categories instead of unrestricted sampling of the web (Asheghi et al., 2016; Pritsos and Stamatatos, 2018). Furthermore, despite the availability of web-scale data in hundreds of languages, until recently, the resources for register identification have focused exclusively on English.

The data for this study consist of four similarly annotated online register collections featuring the CORE corpus series in English (Egbert et al., 2015), Finnish (Laippala et al., 2019), French and Swedish (Repo et al., 2021). All the datasets have been extracted from the unrestricted open web. While the English CORE is extensive, with 34k training examples, the other languages feature merely 2.7–4.6% of that (cf. Table 1).

In this paper, we explore how joint training on the four available CORE corpora can benefit register classification, with a particular interest in improving performance in smaller languages.¹ First, using multilingually pre-trained language models and a custom sampling and training strategy, we compare performance when training on all languages against previous monolingual results on the same corpora, observing gains for the smaller languages. Second, with the aim of creating a universal model fit for all languages, we train a multilingual master model that we evaluate in a zero-shot cross-lingual setting, demonstrating results that land within a relatively short distance from monolingual performances (4–6% F1-score for XLM-R).

2 Related work

Until recently, register identification from the unrestricted web has achieved only modest performance (Sharoff et al., 2010; Asheghi et al., 2014;

¹For code and model, see: <https://github.com/TurkuNLP/multilingual-register-labeling>

Lang.	Train	Dev.	Test	Total
En	33,915	4,845	9,692	48,452
Fi	1,559	222	445	2,226
Fr	909	363	546	1,818
Sv	1,093	435	654	2,182

Table 1: Data set sizes in number of documents.

Biber and Egbert, 2016). Most importantly, the challenges are caused by the range of linguistic variation found on the web. Texts are written without gatekeepers, and not all registers are equally well-defined with discrete class boundaries (Biber and Egbert, 2018; Sharoff, 2018). To this end, Biber and Egbert (2018) suggest to extend the analysis to *hybrid* documents combining characteristics of several register classes, and Sharoff (2018, 2021) examines web genres by prototypical genre classes and text dimensions featuring communicative functions, such as argumentation or reporting.

Despite the difficulty, Laippala et al. (2019) show that multi- and cross-lingual modeling of registers between English and Finnish is possible at practical levels of performance, as they propose a convolutional neural network (CNN) model with multilingual word embeddings to model registers. Further, Repo et al. (2021) demonstrate that pre-trained neural language models, especially XLM-R, can achieve strong performance monolingually on the four aforementioned languages, as well as achieve strong cross-lingual transfer in a zero-shot learning setting from English to other languages.

The benefits of combining several languages during training has been demonstrated for other NLP tasks. Training the multilingual XLM-RoBERTa (XLM-R), Conneau et al. (2020) showed that adding more languages to training leads to better cross-lingual performance on low-resource languages. Comparing the performance of multiple multilingual models across a number of tasks and languages, Hu et al. (2020) noted as well that adding target language data to training provides higher performance. However, they highlighted that a model’s cross-lingual performance varies greatly between languages and tasks – on QA tasks, zero-shot models are very efficient and outperform models trained on 1,000 examples of target-language data. Finally, also the positive effect of sampling under- and overrepresented languages has been demonstrated previously; in

the context of multilingual semantic parsing, Li et al. (2020) perform up- and downsampling of languages based on frequency as part of their sampling strategy, in order to improve multilingual performance.

3 Data

The four datasets we use in this study—CORE, FinCORE, FreCORE and SweCORE—all feature the unrestricted web, however, they have been compiled in different ways. The English CORE is based on unrestricted search queries of extremely frequent n-grams, while the other datasets are randomly sampled from the 2017 CoNLL Shared Task datasets, originally drawn from Common Crawl (Ginter et al., 2017). Table 1 summarizes the data set sizes.

The four datasets have all manual register annotations following the same register taxonomy that was developed during the compilation of the English CORE. The taxonomy is hierarchical, with eight main registers and approximately 30 subclasses, depending on the language-specific version. In this study, we focus on the main register level, which includes the classes *Narrative* (NA), *Informational Description* (IN), *Opinion* (OP), *Interactive Discussion* (ID), *How-to/Instruction* (HI), *Informational Persuasion* (IP), *Lyrical* (LY) and *Spoken* (SP) (for a detailed description, see (Biber and Egbert, 2018)).

In order to reflect the variation found within the data, *hybrid* documents combining characteristics of several registers are also annotated. On the main register level, these display 11–15% of all other language-specific datasets but Finnish. Perhaps because of the different approaches to gathering the corpora, the register distributions differ also for some other classes between CORE and the others. Specifically, the Informational Persuasion class covers only 2.75% of CORE, and 16.82–24.15% of the other datasets, and also the Opinion class covers 16.23% of CORE and 15.23% of FinCORE, but only 6.63% of FreCORE and 6.60% of SweCORE (for details, see Repo et al. (2021)).

4 Methods

4.1 Multilingual language models

We focus on two multilingual deep learning models, namely Multilingual BERT (mBERT, Devlin et al., 2019) and XLM-RoBERTa (XLM-R, (Conneau et al., 2020)), which have been shown

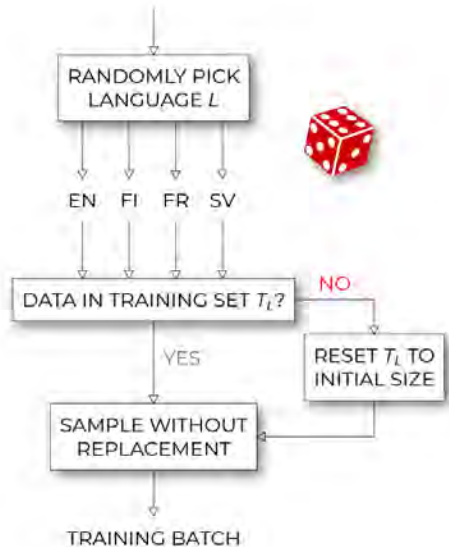


Figure 1: Illustration of the multilingual sampling strategy. Languages are uniformly sampled to generate training batches. A training set is independently reshuffled after a full pass.

to achieve high performance in both monolingual and zero-shot cross-lingual settings of register classification. Repo et al. (2021) show that XLM-R clearly outperforms mBERT by up to 8% points F1-score monolingually and up to 11% points cross-lingually, while both clearly outperform previous state-of-the-art.

Both mBERT and XLM-R are based on the BERT architecture, the first being trained on Wikipedia in 104 languages and the latter on cleaned Common Crawl data in 100 languages. While both models lack an explicit cross-lingual signal, XLM-R has more than double the vocabulary size and was trained on significantly more data for a longer time. We use the large version of XLM-R, whereas mBERT is only available in base size. In various multilingual tasks, XLM-R has been shown to outperform mBERT, which tends to struggle especially with smaller languages such as Finnish and Swedish (Rönnqvist et al., 2019). Nevertheless, we include both models in order to study their relative performances as we introduce a multilingual sampling strategy.

The experiments are performed as multi-label classification in order to support hybrid registers. We use TensorFlow checkpoints of the models through the Huggingface Transformers library and repository (Wolf et al., 2020). We train a deci-

sion layer on top of the top-layer CLS embedding, while also fine-tuning the language model parameters, with a binary cross-entropy loss. The models are evaluated using micro-averaged F1-score and a fixed prediction threshold of 0.5.

4.2 Sampling and training strategy

Since the training sets in the different language corpora we use differ, they risk skewing the class distributions when training on multiple languages at once. In particular, the English set is much larger than the others, and exhibits a somewhat different class distribution (see Section 3, Repo et al. (2021)).

In order to mitigate this problem, we propose a sampling strategy which samples all languages in equal parts during training. The strategy is illustrated in Figure 1. First, for each mini-batch, the language is selected with uniform probability, and then training samples are randomly sampled without replacement. The examples in a language set are reshuffled when they have all been sampled, such that the smaller sets are repeated more often. One training epoch consists of $N \cdot B_1$ mini-batches, where N is the number of languages and B_1 the number of mini-batches in the smallest training set.

In combination with this mode of sampling, we train the models for longer than reported by Repo et al. (2021), typically on the order of 100 epochs, in order to avoid explicitly disregarding any data in the larger training sets. We apply an early stopping criterion on the validation set F1-score, in order to avoid excessive training and to empirically determine when the data sets have been sufficiently repeated. We also use a learning rate about an order of magnitude lower than in the previously reported work to match the longer training.

5 Experiments

We first train models jointly on all four languages following the sampling strategy introduced above, and optimize hyperparameters² for each target language separately, based on development set performance. The optimal model for each language is tested on the respective test set. We compare the multilingual results to the previous state-of-the-art results in monolingual settings, i.e., where one and

²We test learning rates in the range $4e^{-6}$ to $7e^{-5}$ and maximum number of epochs 25 to 175 (affecting rate of warm-up and learning rate decay). Batch size is 7 (capped by available GPU memory) and patience 5 epochs.

mBERT Target	Monolingual (baseline)				Multilingual (ours)				Test diff. F1 (%)
	Dev.		Test		Dev.		Test		
	F1 (%)	Std.	F1 (%)	Std.	F1 (%)	Std.	F1 (%)	Std.	
En	72.80	(0.21)	73.06	(0.09)	68.20	(1.36)	68.63	(1.39)	-4.43
Fi	65.91	(0.85)	64.83	(1.16)	69.25	(1.75)	65.95	(1.06)	1.12
Fr	70.74	(1.67)	68.66	(0.63)	72.49	(0.54)	69.55	(0.36)	0.89
Sv	76.91	(0.45)	76.43	(0.46)	78.49	(0.85)	78.22	(1.17)	1.79
Average excl. En			70.75				70.59		-0.16
			69.97				71.24		0.91
XLM-R									
Target	F1 (%)	Std.	F1 (%)	Std.	F1 (%)	Std.	F1 (%)	Std.	
En	75.80	(0.12)	75.68	(0.05)	72.03	(0.89)	72.43	(0.48)	-3.25
Fi	76.25	(0.45)	73.18	(1.35)	77.53	(0.94)	75.00	(0.53)	1.82
Fr	77.38	(0.51)	76.92	(0.24)	78.72	(0.49)	77.54	(0.99)	0.62
Sv	82.61	(0.37)	83.04	(0.62)	83.92	(0.34)	83.92	(0.34)	0.90
Average excl. En			77.21				77.22		0.01
			77.71				78.82		0.83

Table 2: Performance of models trained in monolingual and multilingual settings, optimized for each language separately. F1-scores are means, N=3.

mBERT Target	Multilingual master model				mBERT Target	Zero-shot, from English (baseline)		Zero-shot, multilingual (ours)	
	Common dev.		Test			Test		Test	
	F1 (%)	Std.	F1 (%)	Std.		F1 (%)	Std.	F1 (%)	Std.
En			66.27	(2.33)	En	–	–	55.15	(2.58)
Fi	71.32	(1.51)	65.27	(1.56)	Fi	50.21	(0.74)	58.46	(0.76)
Fr			69.76	(2.24)	Fr	55.04	(0.66)	62.82	(1.86)
Sv			77.92	(1.21)	Sv	62.53	(0.78)	69.48	(0.72)
Average excl. En			69.81		Average excl. En	–		61.48	
			70.98			55.93		63.59	
XLM-R									
Target	F1 (%)	Std.	F1 (%)	Std.	Target	F1 (%)	Std.	F1 (%)	Std.
En			72.37	(1.17)	En	–	–	63.32	(0.25)
Fi	78.20	(0.04)	75.05	(0.81)	Fi	61.35	(1.26)	69.60	(0.55)
Fr			78.81	(0.89)	Fr	64.27	(1.58)	72.85	(1.74)
Sv			82.36	(0.54)	Sv	69.22	(1.66)	79.49	(0.95)
Average excl. En			77.15		Average excl. En	–		71.31	
			78.74			64.95		73.98	

Table 3: Performance of models validated against a common development set that is balanced between the languages, and tested on the language-specific test sets. F1-scores are means, N=3.

Table 4: Performance of models trained in zero-shot cross-lingual settings, from English to target language (left), and from all other languages to target (right). F1-scores are means, N=3.

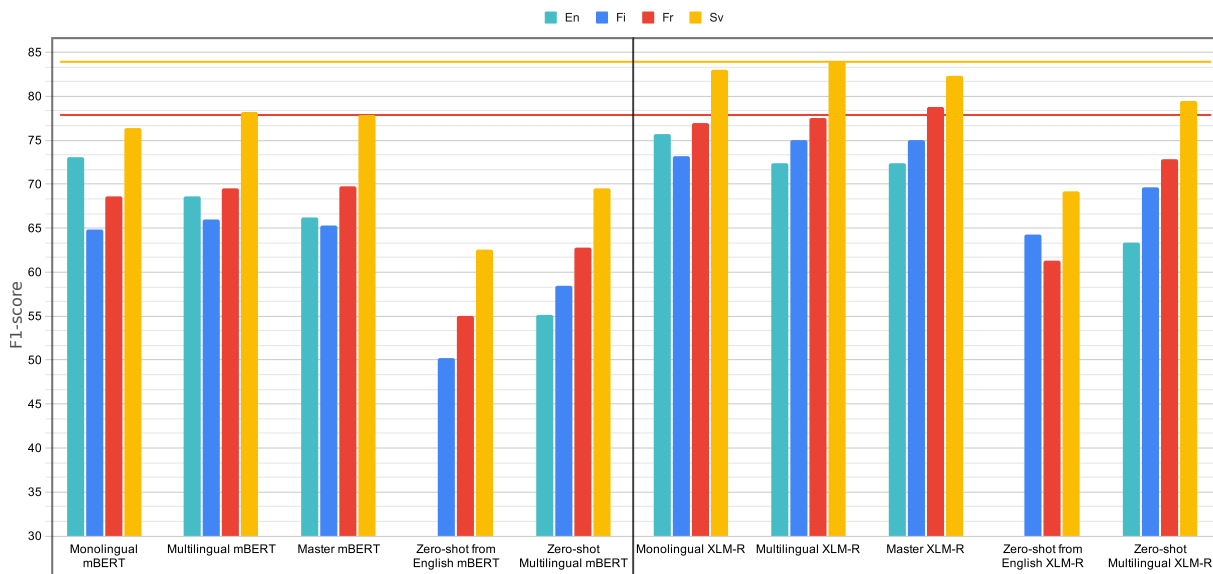


Figure 2: Comparison of all F1-scores. The left box presents the performance of mBERT in the different settings (bar groups) for all languages (color coded) and the right box presents those of XLM-R. Inter-annotator agreement levels (horizontal colored lines) for French and Swedish provide points of reference indicating potential upper bounds for modeling.

the same language is used to train, validate and test the models.

Table 2 presents the results of these experiments (right hand side), as well as the monolingual baseline performances reported by Repo et al. (2021) (left hand side). We observe that both mBERT (above) and XLM-R (below) perform better in multilingual training for all languages except for English. The gains are on average (excluding English) 0.8–0.9% F1-score for the two models, indicating some degree of cross-lingual knowledge transfer from the extra data. Meanwhile, performance for English drops by 3.3–4.4% points, which is likely due to the class distribution being pushed to its disadvantage by the uniform sampling of the otherwise more homogeneous corpora. In terms of average F1-score, the multilingual performance is on par with the previous monolingual models.

Second, after optimizing on each language individually, we perform another hyperparameter search for training a single multilingual model that should favor each language equally, which we call a *master model*. In order to train the master model, we create a common development set based on the individual sets of the languages. The development sets differ in size due to different sizes of the corpora and different data split ratios (see Table 1). We create the common set by upsampling

the Finnish and French and downsampling the English set to the size of the Swedish set; the sets are then concatenated to a total size of 1740. The master model is validated against this set during training. In particular, when to stop training is determined based on the performance on this set, i.e., on the average performance across languages.

Table 3 lists the best performance on the common development set for both mBERT and XLM-R, as well as the performance of both models in each language-specific test set. The level of performance remains stable for the master model, with an average decrease of 0.78% for mBERT and only 0.08% for XLM-R compared to the multilingual results in Table 2.

Third, in order to estimate the performance that can be expected of the master model on an unseen language, we still perform an experiment where each of our four languages is in turn taken as target, and a model is trained with the previously optimized hyperparameters, using the remaining three languages for training and validation (controlling early stopping). The models are tested in each language separately.

The results of this zero-shot cross-lingual experiment are listed in Table 4 (right hand side), along baseline results from previous work studying cross-lingual transfer from English to the other languages (Repo et al., 2021) (left hand side).

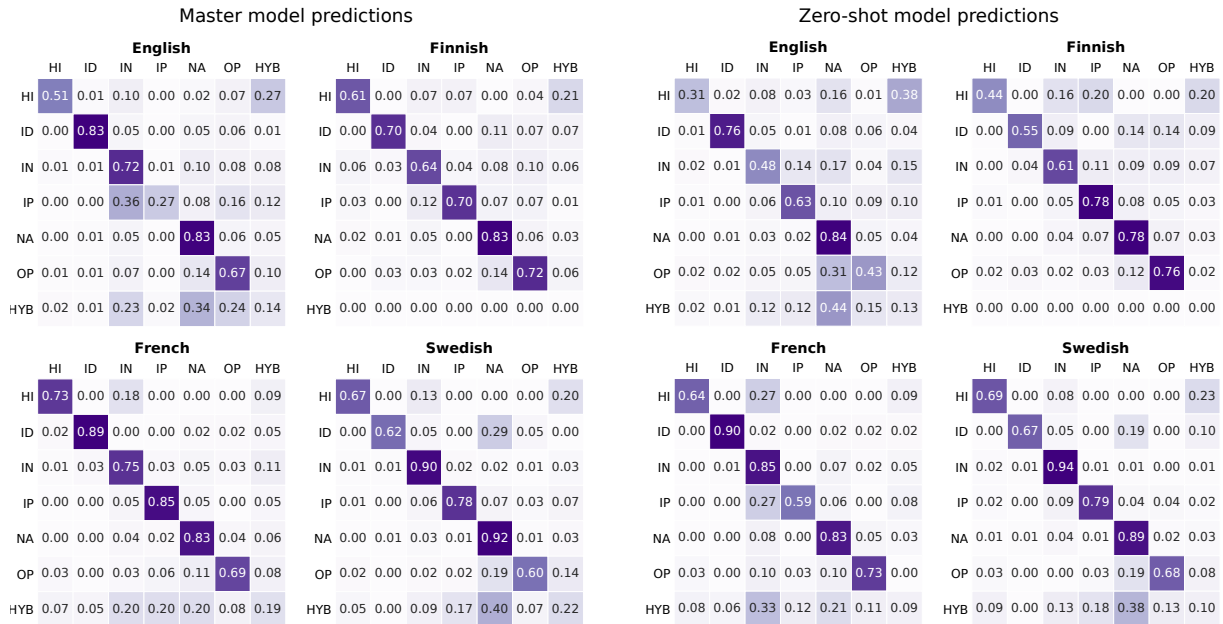


Figure 3: Confusion matrices for predictions in each language using the master multilingual model (left) and the zero-shot cross-lingual models (right). Columns represent predictions and rows true labels for the most common classes, with all hybrid instances represented by HYB only.

The numbers show a significant gain for multilingual modeling over cross-lingual modeling from English only; for mBERT the increase is 5.55% (7.66% excluding English) and for XLM-R 6.36% (9.03%).

Finally, Figure 2 summarizes the F1-scores from the aforementioned tables in a side-by-side comparison. We especially observe how the zero-shot multilingual results take the lead over the baseline of zero-shot from English, in order to approach the levels of the monolingual and multilingual models for which target language is also used for training. The levels of inter-annotator agreement, as reported by Repo et al. (2021), were counted prior to any discussions between the annotators. Although this level should be considered as a lower bound of human agreement, it sets a theoretical boundary for automatic register identification.

6 Error analysis

In order to gain a more detailed understanding of the types of errors the models are making, we study the confusion matrices in Figure 3. These present the correct classifications (diagonal) and misclassifications (rest), both in a single language setting using the master multilingual model and in a cross-lingual setting using the zero-shot model.

The matrices include the six most frequent classes and a separate hybrid class, as the confusions matrix is not defined for the multi-label setting.

We observe that hybrid documents overall are difficult to recognize as such, in particular hybrids composed of Narrative (NA) and another class are often predicted as NA only. Comparing the master model (left) and the zero-shot models (right), we see that the overall patterns are quite similar, while the cross-lingual performance, for instance, in English and Finnish is worse for How-to/Instruction (HI) and Interactive discussion (ID). In Swedish, however, ID performs better cross-lingually, and Swedish generally exhibits the smallest differences between the settings.

Informational description (IN) and Informational persuasion (IP) are difficult to distinguish in English for the master model, whereas the cross-lingual model handles these classes much better, although there is still room for improvement. Distinguishing purely informational texts and those with an intent to persuade is difficult for other zero-shot models as well.

Comparing the cross-lingual matrices with those reported by Repo et al. (2021) for transfer from English to the other languages, we note that our diagonals are significantly crisper, i.e., the classes more frequently correctly predicted. In

their results, especially the classes HI and IP are generally more dispersed, as well as Opinion (OP) for French, NA for Swedish and IN for Finnish (vertically, i.e., other classes are mistaken for IN).

Finally, comparing class-wise F1-score between the master and zero-shot models we observe a 3.1% mean decrease for NA (sd. 1.5%), 5.9% for OP (sd. 2.6%) and 7.6% for IP (sd. 5.4%). Most of the classes are too infrequent in our data for meaningful interpretation of class-wise differences, or the patterns are inconsistent across languages.

7 Discussion

Our results show that multilingual training brings clear advantages to web register identification, in particular for the languages with small amounts of training examples. When allowing training on target data, performance is somewhat improved for these languages, while it remains on par in average. In the zero-shot setting, however, the performance is greatly improved compared to the recent and already strong state-of-the-art results. As illustrated in Figure 2, the multilingual zero-shot XLM-R is closing in on its top-performing counterparts trained monolingually or on all languages.

The fact that the multilingual performance on English is lagging behind is expected, as its class distribution differs notably from that of the other languages, and the uniform sampling is designed to allow the model to learn a mean distribution across the languages. In the zero-shot experiments, the English-targeted model will see relatively little data compared to the other models, which likely works to its disadvantage. In the context of pre-trained language models, English monolingual models are also known to be high-performing; similar results on a multilingual model outperforming other monolingual models but not English have been reported by Hu et al. (2020).

To test how the multilingual model performs in a zero-shot setting, we experimented with a leave-one-out version of the multilingual setting, where a model was trained on all except for the target language data on which the model was tested. Although the results were, as expected, lower than the monolingual and multilingual results where target language was included in training, the gap is closing quickly. With the baseline methods, the average gap between the cross-lingual models and monolingual models has been 12.76% points F1-

score—in our study, it is 3.73% excluding English, 5.9% including English (with XLM-R).

With an average F1-score of 73.98% for Finnish, French and Swedish, we demonstrate that applying this multilingual register classification model in zero-shot settings can be done at very practical levels of performance. This indicates that our multilingual model can be applied without significant loss of accuracy on languages without existing register-annotated corpora, which is an important step toward being able to perform register identification on the truly unrestricted web, also in terms of language.

In particular, these performances are competitive considering the difficulty of the task. As discussed above, the inter-annotator agreements of 78% for French and 84% for Swedish serve as a potential upper bound in modeling. The monolingual models are already very close to this level, and the multilingual zero-shot models are not far.

The competitiveness of multilingual training is particularly interesting in the case of registers. Although the advantages of this multilingual training have been noted before (see Section 2), it is not evident that register identification can benefit from it. Registers are specific to the situation and to the culture where they have been produced. For instance, Opinion blogs can express their points of view differently depending on cultural context, and the level of formality of Speeches and News reports (subregisters of Spoken and Narrative) may vary according to the culture. Also the linguistic means to express functional characteristics associated with registers, such as narration or interaction, differ across languages. These differences can have a drastic effect on the success of the modeling even if the transfer itself works. In the current study, the included languages are all European, which makes also the transfer easier, whereas including more languages and more distant cultures remains a research desideratum.

8 Conclusion

To sum up, our study corroborates the power of multilingual training when modeling registers in languages with a limited amount of training data. We train and make available a multilingual master model for register classification, whose performance is competitive with existing monolingual models. Its zero-shot performance is approaching that of monolingual models, as it improves upon

already strong state-of-the-art results. Considering the estimated level of human agreement on the task, the margin for further improvement is relatively slim. Nevertheless, it is our goal to continue this work in order to achieve robust zero-shot performance in a wide range of languages up to the level of monolingual models. Furthermore, it would be interesting to test the robustness and generalizability of our models by evaluating them against the prototypical web genre categories and Function Text Dimensions presented in (Sharoff, 2018, 2021).

Finally, in the future, we will also investigate register-specific differences in their transfer. Registers differ in terms of how well they are linguistically defined, which naturally also affects their identification (Laippala et al., 2021). For instance, while the linguistic characteristics of many blogs can vary extensively, those of encyclopedia articles remain very similar across texts. This tendency concerns also the cross-lingual similarities of registers, and similarities have already been discovered in particular in the spoken register.

Acknowledgements

We thank the Emil Aaltonen Foundation and Academy of Finland for financial support. We also wish to acknowledge CSC – IT Center for Science, Finland, and the NVIDIA Corporation GPU Grant Program, for computational resources.

References

- Noushin Rezapour Asheghi, Serge Sharoff, and Katja Markert. 2016. Crowdsourcing for web genre annotation. *Language Resources and Evaluation*, 50(3):603–641.
- Rezapour Noushin Asheghi, Katja Markert, and Serge Sharoff. 2014. Semi-supervised graph-based genre classification for web pages. In *Proceedings of TextGraphs-9*, pages 39–47. Association for Computational Linguistics.
- Douglas Biber. 1988. *Variation across speech and writing*. Cambridge University Press.
- Douglas Biber. 2012. Register as a predictor of linguistic variation. *Corpus linguistics and linguistic theory*, 8(1):9–37.
- Douglas Biber and Jesse Egbert. 2016. Using grammatical features for automatic register identification in an unrestricted corpus of documents from the open web. *Journal of Research Design and Statistics in Linguistics and Communication Science*, 2:3–36.
- Douglas Biber and Jesse Egbert. 2018. *Register variation online*. Cambridge University Press.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. <https://doi.org/10.18653/v1/2020.acl-main.747> Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Jesse Egbert, Douglas Biber, and Mark Davies. 2015. Developing a bottom-up, user-based method of web register classification. *Journal of the Association for Information Science and Technology*, 66:1817–1831.
- Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. <http://hdl.handle.net/11234/1-1989> CoNLL 2017 shared task - Automatically annotated raw texts and word embeddings. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. <http://arxiv.org/abs/2003.11080> Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization.
- Veronika Laippala, Jesse Egbert, Douglas Biber, and Aki-Juhani Kyröläinen. 2021. Exploring the role of lexis and grammar for the stable identification of register in an unrestricted corpus of web documents. *Lang Resources Evaluation*.
- Veronika Laippala, Roosa Kyllönen, Jesse Egbert, Douglas Biber, and Sampo Pyysalo. 2019. <https://www.aclweb.org/anthology/W19-6130> Toward multilingual identification of online registers. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 292–297. Linköping University Electronic Press.
- Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2020. <http://arxiv.org/abs/2008.09335> Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark.

- Anuj Mahajan, Sharmistha Jat, and Shourya Roy. 2015. <https://doi.org/10.18653/v1/K15-1034> Feature selection for short text classification using wavelet packet transform. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 321–326. Association for Computational Linguistics.
- Dimitrios Pritsos and Efstathios Stamatatos. 2018. <https://doi.org/10.1007/s10579-018-9418-y> Open set evaluation of web genre identification. *Language Resources and Evaluation*, 52(4):949–968.
- Liina Repo, Valtteri Skantsi, Samuel Rönnqvist, Saara Hellström, Miika Oinonen, Anna Salmela, Douglas Biber, Jesse Egbert, Sampo Pyysalo, and Veronika Laippala. 2021. Beyond the english web: Zero-shot cross-lingual and lightweight monolingual classification of registers. In *Proceedings of the EACL 2021 Student Research Workshop*.
- Samuel Rönnqvist, Jenna Kanerva, Tapio Salakoski, and Filip Ginter. 2019. Is multilingual BERT fluent in language generation? In *Proceedings of the 1st NLPL Workshop on Deep Learning for Natural Language Processing*.
- Serge Sharoff. 2018. Functional text dimensions for the annotation of web corpora. *Corpora*, 1(13):65–95.
- Serge Sharoff. 2021. Genre annotation for the web: text-external and text-internal perspectives. *Register Studies*.
- Serge Sharoff, Zhili Wu, and Katja Markert. 2010. The web library of babel: evaluating genre collections. In *Proceedings of LREC*.
- Bonnie Webber. 2009. Genre distinctions for discourse in the Penn TreeBank. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 674–682. Association for Computational Linguistics.
- Marlies Van der Wees, Arianna Bisazza, and Christof Monz. 2018. Evaluation of machine translation performance across multiple genres and languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA).
- Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

Neural Morphology Dataset and Models for Multiple Languages, from the Large to the Endangered

Mika Hämäläinen, Niko Partanen, Jack Rueter and Khalid Alnajjar

Faculty of Arts

University of Helsinki

firstname.lastname@helsinki.fi

Abstract

We train neural models for morphological analysis, generation and lemmatization for morphologically rich languages. We present a method for automatically extracting substantially large amount of training data from FSTs for 22 languages, out of which 17 are endangered. The neural models follow the same tagset as the FSTs in order to make it possible to use them as fallback systems together with the FSTs. The source code¹, models² and datasets³ have been released on Zenodo.

1 Introduction

Morphology is a powerful tool for languages to form new words out of existing ones through inflection, derivation and compounding. It is also a compact way of packing a whole lot of information into a single word such as in the case of the Finnish word *hatussanikinko* (in my hat as well?). This complexity, however, poses challenges for NLP systems, and in the work concerning endangered languages, morphology is one of the first NLP problems people address.

The GiellaLT infrastructure (Moshagen et al., 2014) has HFST-based (Lindén et al., 2013) finite-state transducers (FSTs) for several morphologically rich (and mostly Uralic) languages. These FSTs are capable of lemmatization, morphological analysis and morphological generation of different words.

These transducers are at the core of this infrastructure, and they are in use in many higher level NLP tasks, such as rule-based (Trosterud, 2004) and neural disambiguation (Ens et al., 2019), dependency parsing (Antonsen et al., 2010) and

machine translation (Pirinen et al., 2017). The transducers are also in constant use in several real world applications such as online dictionaries (Rueter and Hämäläinen, 2019), spell checkers (Trosterud and Moshagen, 2021), online creative writing tools (Hämäläinen, 2018), automated news generation (Alnajjar et al., 2019), language learning tools (Antonsen and Argese, 2018) and documentation of endangered languages (Gerstenberger et al., 2017; Wilbur, 2018). As an additional important application we can mention the wide use of FSTs in the creation of Universal Dependencies treebanks for low-resource languages, at least with Erzya (Rueter and Tyers, 2018), Northern Saami (Tyers and Sheyanova, 2017) Karelian (Pirinen, 2019a) and Komi-Zyrian (Partanen et al., 2018).

Especially in the context of endangered languages, accuracy is a virtue. Rule-based methods not only serve as NLP tools but also as a way of documenting languages in a machine-readable fashion. Members of language communities do not benefit, for example, from a neural spell checker that works to a degree in a closed test set, but fails miserably in real world usage. On the contrary, a rule based description of morphology can only go so far. New words appear and disappear all the time in a language, and keeping up with that pace is a never ending job. This is where neural models come in as they can learn to generalize rules for out-of-vocabulary words as well. Pirinen (2019b) also showed recently that at least with Finnish the neural models do outperform the rule-based models. This said, Finnish is already a larger language, so the experience doesn't necessarily translate into low-resource scenario (see Hämäläinen 2021).

The purpose of this paper is to propose neural models for the three different tasks the GiellaLT FSTs can handle: morphological analysis (i.e. given a form such as *kissan*, produce the

¹<https://github.com/mikahama/uralicNLP/wiki/Neural-morphology>

²<http://doi.org/10.5281/zenodo.3926769>

³<http://doi.org/10.5281/zenodo.3928628>

morphological reading $+N+Sg+Gen$), morphological generation (i.e. given a lemma and a morphology, generate the desired form such as $kissa+N+Sg+Gen$ to $kissan$) and lemmatization (i.e. given a form, produce the lemma such as $kissan$ to $kissa$ ‘a cat’). The goal is not to replace the FSTs, but to produce neural fallback models that can be used for words an FST does not cover. This way, the mistakes of the neural models can easily be fixed by fixing the FST, while the overall coverage of the system increases by the fact that a neural model can cover for an FST.

The main goal of this paper is not to propose a state of the art solution in neural morphology. The goal is to first build the resources needed to train such neural models so that they will follow the same morphological tags as the GiellaLT FSTs, and secondly train models that can be used together with the FSTs. All of the trained models will be made publicly available in a Python library that supports the use of the neural models and the FSTs simultaneously. The dataset built in this paper and the exact train, validation and test splits used in this paper have been made publicly available for others to use on the permanent archiving platform Zenodo.

2 Constructing the Dataset

We are well aware of the existence of the popular UniMorph dataset (McCarthy et al., 2020). However, it does not suit our needs of two reasons. One reason is the incompatible morphological tagset. Our goal is to build models that can directly be used side-by-side with the existing FSTs, which means that the data has to follow the same formalism. Conversion is not a possibility, as the main reason we are not interested in using the UniMorph data is its limited scope; not only does it not cover all the languages we are dealing with in this paper, but it does not cover any cases of complex morphology. For example, the Finnish dataset does not cover possessive suffixes, question markers, comparative, superlative etc. Such a data would not be on par with the output produced by the FSTs.

We produce the data for the following languages: German (deu), Kven (fkv), Komi-Zyrian (kpv), Mokhsa (mdf), Mansi (mns), Erzya (myv), Norwegian Bokmål (nob), Russian (rus), South Sami (sma), Lule Sami (smj), Skolt Sami (sms), Võro (vro), Finnish (fin), Komi-Permyak (koi),

Latvian (lav), Eastern Mari (mhr), Western Mari (mrj), Namonuito (nmt), Olonets-Karelian (olo), Pite Sami (sje), Northern Sami (sme), Inari Sami (smn) and Udmurt (udm). A vast majority of these languages are greatly endangered (Moseley, 2010).

We use the FSTs and dictionaries from the GiellaLT with the UralicNLP (Hämäläinen, 2019) library to build the datasets for training the models. We do this in a clever way by taking all open class part-of-speech words from the dictionaries for each language and use the FSTs to produce all morphological readings for them. The number of words in the GiellaLT dictionaries is shown in Table 1. The FSTs do not let us do this by default, so we build a regular expression transducer that finds all possibilities for an input word and its part-of-speech. In order to build the regular expression, we query all alphabets in the transducer that contain one of the following strings for exclusion: *#*, *Der*, *Cmp* or *Err*. This will remove compounds, erroneously spelled forms and derivations. Derivations need to be excluded because otherwise the transducers would produce derivations of derivations and so on. Once the regular expression transducer is composed with the FST analyzer, we can use HFST to extract the transducer paths to get a list of all the possible morphological forms of the input word. From these, we filter out *Clt* and *Foc* tags because these multiply the number of possible morphological forms, especially since multiple different clitics can be appended after each other, and some times even in multiple different orders. We also remove tags indicating non-standard forms, *Use* and *Dial*, and *Sem* tags that are used in language learning tools as well as contextual disambiguation to categorize semantically similar words. Table 2 shows how many unique inflectional forms each part-of-speech category has per language.

We use the method described above to produce the data with all the open class part-of-speech words in the GiellaLT dictionaries for each language. For languages with bigger dictionaries, the maximum number of lemmas used per part of speech is set to 2100, in which case the lemmas are also picked at random. We use the typical split ratio and split 70% of the data for training, 15% for validation and 15% for testing. The split is done on the lemma level and for each part-of-speech separately. This means that the test and valida-

	deu	fin	fkv	koi	kpv	lav	mdf	mhr	mns	mrj	myv	nob	olo	rus	sje	sma	sme	smj	smn	sms	udm	vro
N	8741	51916	5936	558	20042	9738	17196	14079	2263	2529	10234	32009	5942	24691	2685	5946	37943	4331	13826	21158	10722	4703
Adv	588	6036	652	89	2942	953	1771	2346	-	444	743	1743	14	2546	-	543	1314	343	1146	1729	985	122
V	4021	27875	1445	532	12504	2601	11983	9954	4924	2456	3781	7432	2782	14348	1751	5208	7724	3130	5436	5033	3669	4129
A	2768	13056	917	128	5218	1652	4407	5116	-	1031	2926	3236	2134	11054	185	645	2927	468	2295	3898	1550	1019

Table 1: The sizes of the GiellaLT dictionaries per part-of-speech

	deu	fin	fkv	koi	kpv	lav	mdf	mhr	mns	mrj	myv	nob	olo	rus	sje	sma	sme	smj	smn	sms	udm	vro
N	24	850	50	788	183	24	83	208	151	162	19	17	98	75	16	50	727	297	496	339	744	26
Adv	1	16	1	2	4	-	4	3	-	2	2	2	-	1	-	3	8	3	2	3	1	6
V	254	6667	139	198	249	1245	894	59	-	40	10	21	726	693	38	58	302	144	382	177	156	119
A	150	1244	77	4	244	44	127	4	-	2	5	15	217	39	52	75	1347	187	100	627	54	100

Table 2: Number of unique inflectional forms per part-of-speech category

tion sets will consist exclusively of out of vocabulary words that have not appeared in the training in any inflectional form. This also means that the ratios are the same for each part-of-speech, 70% of the adjectives are used in the training, 70% of the verbs and so on. The actual sizes can be seen in Table 3.

The reason why we do the testing purely on out-of-vocabulary words is simply to test the accuracy of the models in the scenario that is more close to the one they are trained for, namely, in cases where the FSTs fail in their coverage.

3 Experiments and Results

In this section, we cover the neural architecture for the three separate morphological tasks: lemmatization, analysis and generation. We also show the results of the models in these tasks for each language, and present an error analysis on the Finnish and Komi-Zyrian by taking a closer look at the results.

3.1 The Neural Model

Over recent years, there has been a growing body of work on different neural approaches for low resourced languages in morphological analysis (Moeller et al., 2019; Schwartz et al., 2019), lemmatization (Kondratyuk, 2019; Silfverberg and Tyers, 2019) and generation (Oseki et al., 2019; Yu et al., 2020). Most notably the use of bi-directional LSTM architecture seems to be supported by most of the recent related work for analysis, generation and lemmatization.

It is important to note that we approach the lemmatization and analysis from the same point of view as the FSTs. This means that it is a strictly morphological process, and the question of disambiguation is left for another part of the GiellaLT NLP pipeline, namely constraint grammar rules

(Bick and Didriksen, 2015). There is a plethora of work dealing with in-context lemmatization (Manjavacas et al., 2019; Malaviya et al., 2019), morphological analysis (Lim et al., 2018; Zalmout and Habash, 2020) and part-of-speech tagging (Perl et al., 2020; Hoya Quecedo et al., 2020), but that is not what we are aiming for. We are aiming for neural models that can be used to complement the already existing systems relying on the GiellaLT infrastructure.

For all three tasks, we train a character based bi-directional LSTM model (Hochreiter and Schmidhuber, 1997) by using OpenNMT-py (Klein et al., 2017) with the default settings except for the encoder where we use a BRNN (bi-directional recurrent neural network) (Schuster and Paliwal, 1997) instead of the default RNN (recurrent neural network) as BRNN has been shown to provide a performance gain in a variety of tasks. We use the default of two layers for both the encoder and the decoder and the default attention model, which is the general global attention presented by Luong et al. (Luong et al., 2015).

Table 4 shows an example of the input and output of the training data in each of the three different tasks. Words are split into characters on both the input and output side of the data. Different morphological tags are treated as separate tokens, this means that FST morphologies consisting of multiple tags such as *N+Msc+Sg+Dat* are simply split by the plus sign. We train a separate model for each task, meaning that we train three different models for each language: one for lemmatization, analysis and generation. All models have shared the same random seed (3435), therefore training the models again with this seed should result in the exact same results we are reporting in this paper.

	deu	fin	fkv	koi	kpv	lav	mdf	mhr	mns	mrj	myv	nob	olo	rus	sje	sma	sme	smj	smn	sms	udm	vro
train	394k	14486k	286k	483k	873k	320k	1267k	666k	283k	232k	45k	37k	1054k	243k	80k	108k	799k	648k	1167k	2831k	943k	257k
val	87k	3061k	62k	105k	186k	68k	276k	142k	60k	50k	9k	8k	229k	51k	17k	22k	177k	145k	249k	628k	202k	54k
test	84k	3109k	60k	105k	186k	68k	274k	142k	60k	50k	9k	8k	221k	53k	16k	23k	179k	143k	253k	624k	203k	55k

Table 3: Sizes of the datasets for each language. The splits do not share vocabulary.

	input	output
lemmatization	k a u n i i m p a n s a k o	k a u n i s
analysis	k a u n i i m p a n s a k o	A Comp Sg Gen PxSg3 Qst
generation	k a u n i s A Comp Sg Gen PxSg3 Qst	k a u n i i m p a n s a k o

Table 4: Example of the training data for each task

3.2 Results

We report the performance of the models in terms of accuracy, meaning how many results were fully right (entirely correct lemma, entirely correctly generated form and entirely correct morphological analysis). In addition, we report CER (character error rate) for the lemmatizers and generators, and a MER (morphological error rate) for the analyzers. These values indicate how close the model got to the correct result even if some of the results were a bit erroneous.

The results can be seen in Table 5, the models reaching to an accuracy to over 80 % are highlighted in bold. The results indicate that lemmatization is the easiest task for the model to learn, and after that generation. Morphological analysis is the most difficult task as it receives the scores lower than the generation or lemmatization. Needless to say, some results are exceptionally good for specific languages such as for Erzya (myv) and Western Mari (mrj), while they are not good for others like Finnish (fin) and German (deu). This calls for more investigation of the results.

Figure 1 shows the accuracy of each model based on the morphological complexity of the input. The complexity is measured by the number of morphological tags in the FST produced data. The complexity axis of the plots shows a relative complexity for each language, meaning that 1.0 has the maximum number of tags, 0.8 shows results for input having 80% of the maximum number of tags and so on. The maximum complexity is shown in brackets after the language ISO-code. Analyzers seem to have a lower accuracy for most of the languages when the complexity is small. This is probably due to the fact that shorter word forms tend to have more ambiguity to begin with and might be analyzed as a word different from the one in the gold standard. For many languages, the accuracy

increases towards the average complexity and drop again for the most complex forms. It is to be remembered that these accuracies are also affected by the peculiarities of the transducers themselves and their tagging conventions.

Lemmatizers seem to follow the pattern of the analyzers but do so more clearly. Lemmatization of morphologically simple forms is not as easy as more complex forms. However, as the complexity increases, the lemmatization accuracy does not drop for most of the languages. This has probably something to do with the fact that unlike morphological tags, the word forms follow clearer patterns as they do not have such a large amount of subjectivity in the tagging decisions the different linguists working on these transducers have introduced.

Generators are very even for most of the languages in the sense that they produce consistently around the same accuracy regardless of the morphological complexity. Although, some of the languages follow a more analyzer like pattern, generating wrong with small and large morphological complexity.

Table 6 shows the most difficult tags for the analyzers. The missing predictions column shows the most frequent tags the analyzer did not predict even though they were in the gold data, and the wrong predictions column shows the most frequent ones the analyzer predicted but were not in the gold data. We can see that many of the most challenging tags are shared by different languages. In various Uralic languages, for example, connegatives and imperatives, or connegatives and infinitives, are homonymous, and cannot be predicted correctly just from the surface form alone. Similarly cases such as illative and inessive are in many complex forms homonymous in Permic languages, which surfaces in missing pre-

	deu	fin	fkv	koi	kpv	lav	mdf	mhr	mns	mrj	myv	nob	olo	rus	sje	sma	sme	smj	smn	sms	udm	vro
gen acc	0,65	0,64	0,68	0,67	0,78	0,95	0,85	0,58	0,78	0,90	0,93	0,94	0,83	0,97	0,77	0,69	0,73	0,67	0,57	0,40	0,87	0,82
gen CER	5,61	8,03	3,70	8,67	3,75	1,21	1,77	11,76	3,92	1,77	0,67	1,23	2,12	0,48	4,28	5,81	4,19	3,54	4,25	6,65	1,90	3,35
lem acc	0,88	0,68	0,80	0,70	0,87	0,85	0,93	0,88	0,79	0,88	0,90	0,76	0,87	0,82	0,72	0,71	0,06	0,70	0,67	0,79	0,92	0,79
lem CER	2,71	12,37	5,85	11,41	1,21	4,34	1,11	2,46	4,87	3,82	1,50	5,71	3,76	4,25	6,78	6,96	55,72	9,13	7,78	4,45	2,75	5,81
ana acc	0,11	0,57	0,86	0,78	0,88	0,39	0,61	0,94	0,77	0,92	0,98	0,49	0,86	0,36	0,73	0,60	0,56	0,53	0,42	0,42	0,76	0,74
ana MER	35,40	16,66	6,52	7,24	3,06	18,24	11,54	7,76	4,75	0,41	0,41	38,09	5,85	19,04	19,45	22,91	17,24	22,48	23,20	20,11	10,90	9,82

Table 5: Results of the models for different languages on out-of-vocabulary data

	Missing predictions	Wrong predictions
deu	Def, Pl, Acc, Dat, Neu, Gen, Msc, Fem, NoArt, Nom	NoArt, Fem, Msc, Indef, Sg, Gen, Acc, Nom, Def, Neu
fin	PxSg3, A, PxPl3, N, Sg, Pl, Nom, Gen, Par, Pss	V, Act, PxPl3, PrfPrc, PxSg3, Ind, PrsPrc, Pss, Prs, Sg
fkv	A, Act, N, Sg, V, Pl, Ind, Inf3, Nom, Pl3	N, Act, A, V, Sg, Pl, Ind, Pass, Prs, Inf3
koi	IV, TV, AprIne, AprIll, Ill, Prs, V, So/CP, Apr, Ind	Apr, So/CP, TV, Ine, AprIne, Fut, Sg, N, IV, Nom
kpv	Ine, TV, IV, Fut, Prs, Ill, V, Pl3, Sg1, PxSg1	Ill, IV, TV, Prs, Ine, Fut, Sg, Sg3, N, Pl1
lav	IV, Fem, Acc, Pl, Sg, Nom, TV, Voc, Def, Gen	TV, Gen, Msc, Sg, Pl, Indef, Loc, IV, Fem, Acc
mdf	IV, TV, V, Ind, Prt2, OcPl3, N, A, PxSg2, NomAct	TV, IV, N, Conj, OcSg3, Def, A, V, OcPl1, ScSg1
mhr	N, Sg, V, So/CP, Nom, Ill, Ind, So/PC, Gen, Ger	V, Ind, N, Sg, Adv, Nom, Prs, Sg3, Imprt, A
mns	PxPl2, Sg, Pl, PxDu2, Nom, PxSg2, Du, Lat, Abl, Loc	PxDu2, Pl, Sg, PxSg2, Nom, PxPl3, Du, Lat, Tra, PxPl2
mrj	Sg, N, Lat, Prs, V, Nom, Ind, Imprt, Ine, PxPl3	Prt1, Ind, V, Ill, N, Sg, Nom, Sg3, Ine, Prs
myv	N, A, Tra, Ela, Abl, Ine, Interr, PxSg2	A, N, Abl, Ine, Ela, Ill, Tra, NomAg, V, IV
nob	A, Pos, Indef, Sg, Pl, PrfPrc, Def, Fem, V, MF	Msc, Ind, V, Sg, Indef, Prt, Neu, Def, N, Pl
olo	Ins, N, A, ConNeg, V, Sg, Act, PrfPrc, Nom, Pl	Gen, ConNeg, Act, V, N, Ind, A, Sg, Prs, PrsPrc
rus	TV, Acc, Neu, Gen, Anim, Inan, Impf, Dat, Pass, IV	IV, Loc, Msc, AnIn, Nom, Perf, TV, Acc, Gen, Sg
sje	Sg, V, Com, Prs, N, Sg2, Pl, Gen, Ind, Nom	N, Sg, Pl, V, Ind, Nom, Prs, Prt, Gen, Ine
sma	IV, TV, N, Ind, V, Ess, Sg, Pl, A, Prs	Sg, IV, TV, N, Ind, Com, Prs, Pl, Nom, Ill
sme	Acc, Gen, TV, Sg, Pl, Loc, IV, N, Com, Nom	Gen, Sg, Acc, IV, TV, Nom, Pl, Com, Loc, A
smj	Com, Sg, N, Pl, PxDu2, PxDu1, Gen, Acc, IV, NomAg	Gen, TV, V, PxPl2, PxPl1, Pl, Sg, Nom, IV, Ine
smn	Acc, Sg, Gen, PxPl2, PxSg3, N, PxPl3, IV, PxPl1, TV	PxDu3, Gen, PxDu2, Nom, A, Ill, Sg, PxSg1, PxSg2, Pl
sms	N, Sg, V, Acc, Pl, Gen, Ill, Com, Ind, Nom	A, Pl, Nom, Sg, Gen, Loc, Acc, Com, Ill, Par
udm	Ill, N, Opt, Sg, ConNeg, Ind, PxSg3, Imprt, Sg2, Ine	ConNeg, Ind, Ine, Sg3, Fut, Nom, N, A, Det, Sg
vro	A, Pss, Act, V, Pl, Sg, Ind, N, Gen, ConNegII	Sg, Nom, Act, N, A, Pl, Sg1, Prs, Ind, Par

Table 6: The top 10 most difficult tags for the analyzers

dictions of all these languages. In the languages where transitivity is a feature coded into FST, there are regular problems in predicting these categories correctly. Similarly, in many Indo-European languages gender is primarily a lexical category, and in many instances the model cannot predict it correctly in cases where only the surface form that doesn't show the gender is presented. In the Section 3.3 we go through more in detail this kind of instances, for example, in relation to purely lexically determined Komi-Zyrian stem consonants.

Table 7 shows the morphological constructions that were the most difficult ones for the models to lemmatize and generate correctly in their respective columns. For instance, the Erzya (myv) generation indicates the translative with subsequent possessive-suffix marking is the most problematic. If it had been lemmatization, the explanation would point to the extreme infrequency of these translative forms and the fact that there is an ambiguity with genitive and nominative forms of derivations in *ks*. Lemmatization for Erzya, however, appears to have no issues with ambiguity at all. The same difficulties are not shared by

other languages, but seem to all be language specific. Eastern and Meadow Mari (mhr), for example, appear to have difficulties with generation and lemmatization of nearly the same tag set, namely, the illative plural with a third person plural possessive suffix (ordered: possessive, plural and finally case marker). Looking at the sibling language Western Mari (mrj), we will note that there is a different tagging strategy in use, but here as well there seems to be an intersection where the same forms present problems for both generation and lemmatization.

This could be seen as a type of sanity test whereby simple flaws in the transducers might be detected. The Latvian (lav) transducer is a blatant example of inconsistencies in transducer development. The problem, which has now been addressed and corrected, was in the multiple exponence of part-of-speech tags, i.e. there are double +V and +N tags due to the introduction of automated part-of-speech tagging in XML dictionary to FST formalism transformation without removing the part-of-speech tagging in subsequent continuation lexica of the rule-based transducer.

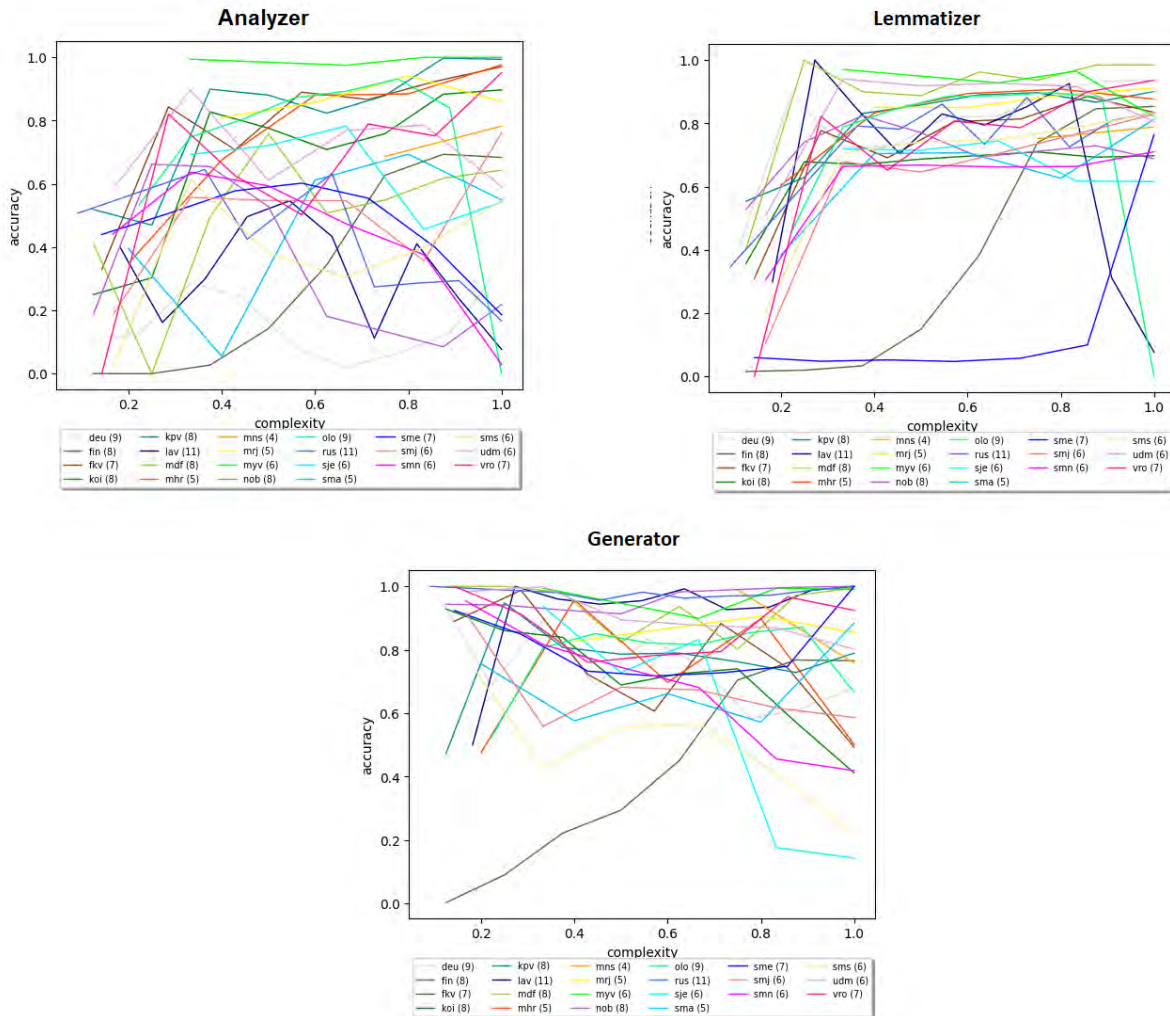


Figure 1: Accuracies based on morphological complexity

Development of the Mari pair might be greatly enhanced through the introduction of a segment-ordering tag in Western or Hill Mari (mrj), which would bring it closer to the strategy followed in the Eastern and Meadow Mari (mhr) use of +So/PNC. These questions with tag and suffix ordering appear also as important factor in Komi-Zyrian morphological generation, as discussed in Section 3.3.

3.3 Error Analysis

In this section, we take a closer look at the result of the Finnish (fin) and Komi-Zyrian (kpv) models in order to better understand their shortcomings.

3.3.1 Finnish

For lemmatization Finnish offered one of the worst results, which makes it an interesting target for error analysis. Some of the obvious errors are re-

lated to extremely common word formation patterns, which the model for some reason is not able to generalize. One of these pattern belongs to adjectives and nouns formed with suffix -inen, for example *pienimuotoisissani* ‘in my most minor (things)’ the correct lemmatization would be *pienimuotoinen*, but the model returns *pienimuotoida*, which doesn’t mean anything. Interestingly, it gives very consistently similar forms to different variants of the same word, so the model appears to believe this is the correct lemma. We can analyze that out of all Finnish lemmatization errors -inen derivations are involved in 7.7% of all mistakes. Thereby future work should investigate what can cause such a gap in the models prediction abilities, as impact in this can lead into rapid improvements. One phenomena we observed is that Finnish FST also produces incorrect forms, such as *pienimuo-*

	generator	lemmatizer
deu	V+PrfPrc+Pos+Pl+Nom+Indef, V+PrfPrc+Pos+Fem+Sg+Nom+Indef, V+PrfPrc+Pos+Pred, V+PrfPrc+Pos+Fem+Sg+Acc+Indef, V+PrfPrc+Pos+Neu+Sg+Acc+Def	N+Msc+Pl+Dat, N+Msc+Pl+Gen, N+Msc+Pl+Nom, N+Msc+Pl+Acc, N+Msc+Sg+Dat
fin	A+Sg+Ess+PxSg3, A+Sg+Ess+PxPl3, A+Sg+Ess+PxPl3+Qst, A+Sg+Ess+PxSg3+Qst, N+Pl+Par+PxPl3+Qst	A+Sg+Ess+PxSg3, A+Sg+Ess+PxPl3, A+Sg+Ess+PxPl3+Qst, A+Sg+Ess+PxSg3+Qst, N+Pl+Par+PxSg3
fkv	V+Act+Inf3+A+Pl+Superl+Par, A+Pl+Superl+Par, N+Pl+All, V+Act+Inf3+A+Pl+Par, V+Act+Inf3+A+Pl+Gen	N+Pl+All, N+Pl+Par, N+Pl+Gen, N+Sg+Par, N+Pl+Abe
koi	V+Ind+Prt2+Pl3+Comp, V+Ind+Prt2+Pl3, V+IV+Ind+Prt2+Pl3+Comp, V+IV+Ind+Prt2+Pl3, V+TV+Ind+Prt2+Pl3	N+Sg+Ela+Comp+Cop+Pl, N+Sg+Ine+PxPl1+Comp+Cop+Pl, N+Sg+Ine+PxPl2+Comp+Cop+Pl, N+Sg+Ine+PxPl3+Comp+Cop+Pl, N+Sg+Ela+PxSg1+Comp+Cop+Pl
kpv	N+Sg+Com+PxSg2, N+Sg+Com+PxSg3, N+Sg+Egr+PxSg1+Comp, N+Sg+Egr+PxSg1, N+Sg+Egr+PxSg1+Comp+Cop+Pl	N+Sg+Acc, Adv, N+Sg+Prt+PxPl1, N+Sg+Com+PxSg2, N+Sg+Com+PxSg3
lav	V+V+TV+PrsPrc+Act+Msc+Sg+Voc+Def, V+V+TV+PrsPrc+Pss+Msc+Sg+Voc+Def, V+V+TV+PrfPrc+Pss+Msc+Sg+Voc+Def, N+N+Msc+Sg+Voc, V+V+IV+PrsPrc+Act+Msc+Sg+Voc+Def	N+N+Msc+Sg+Voc, N+N+Fem+Sg+Voc, N+N+Fem+Pl+Gen, N+N+Fem+Sg+Acc, N+N+Fem+Sg+Loc
mdf	V+IV+NomAg+Pl+Gen+PxSg3, V+IV+NomAg+Pl+Nom+PxSg3, V+IV+NomAg+SP+Cau+Indef, V+IV+NomAg+Sg+Dat+PxSg1, V+IV+NomAg+Pl+Dat+PxSg3	N+Sg+Nom+Indef, N+SP+Gen+Indef, N+Sg+Dat+PxSg2, N+Sg+Dat+PxSg1, N+SP+Tra+Indef
mhr	N+Pl+III+PxSg3+So/PNC, N+Pl+III+PxPl1+So/PNC, N+Pl+III+PxPl2+So/PNC, N+Pl+III+PxPl3+So/PNC, N+Pl+III+PxSg1+So/PNC	Adv, N+Pl+III+PxSg3+So/PNC, N+Pl+III+PxSg2+So/PNC, N+Sg+III+PxSg3+So/CP, N+Pl+III+PxSg1+So/PNC
mms	N+Du+PxDu1+Abl, N+Du+PxDu1+Ins, N+Du+PxDu1+Nom, N+Du+PxDu1+Loc, N+Du+PxDu1+Lat	N+Pl+PxSg2+Ins, N+Pl+PxSg2+Loc, N+Pl+PxSg2+Abl, N+Pl+PxSg2+Nom, N+Pl+PxSg2+Lat
mrj	N+Sg+Ine+PxSg3, N+Sg+III, N+PxSg2+Pl+III, N+Sg+PxSg2+III, N+PxSg1+Pl+Lat	N+Sg+III, N+Sg+Gen, N+Sg+Acc, N+Sg+Nom, N+Sg+Ine+PxSg3
myv	N+SP+Tra+PxSg2, N+SP+Tra+PxPl1, N+SP+Tra+PxPl2, N+SP+Tra+PxPl3, N+SP+Tra+PxSg3	V+IV+Act+PrsPrc, V+IV+NomAg+SP+III+PxSg2, V+IV+NomAg+SP+III+PxPl1, V+IV+NomAg+SP+III+PxSg3, V+IV+NomAg+SP+Ine+PxPl1
nob	N+Neu+Pl+Def, V+Ind+Prt, N+Neu+Pl+Indef, V+PrfPrc, A+Superl+Def	V+Imp, A+Pos+Neu+Sg+Indef, A+Pos+Fem+Sg+Indef, V+Ind+Prt, A+Pos+Msc+Sg+Indef
olo	V+Act+PrsPrc+Pl+Abe, V+Act+PrsPrc+Pl+Abe+Qst, N+Pl+Abe, N+Pl+Abe+Qst, N+Sg+Abe+Qst	N+Sg+Nom, N+Sg+Nom+Qst, N+Sg+Abe, N+Pl+Abe+Qst, N+Sg+Abe+Qst
rus	V+Perf+IV+Imp+Pl2, V+Perf+IV+Imp+Sg2, V+Perf+IV+Fut+Sg3, V+Perf+IV+Fut+Sg2, V+Perf+IV+Fut+Sg1	Adv, A+Neu+Sg+Pred, A+Msc+Sg+Pred, V+Perf+IV+Imp+Sg2, A+Msc+AnIn+Sg+Loc
sje	N+Pl+Ela, N+Pl+Com, N+Sg+Ela, N+Sg+Com, V+Pot+Sg3	N+Pl+Com, N+Pl+Ela, N+Sg+Ela, N+Sg+Com, V+Ind+Prs+Sg3
sma	N+Pl+Gen, N+Pl+Com, N+Sg+Com, N+Pl+III, N+Ess	N+Pl+Gen, N+Sg+Gen, N+Ess, N+Pl+Ine, N+Pl+Nom
sme	A+Comp+Sg+Nom+Qst, A+Comp+Sg+Nom, A+Comp+Attr, A+Comp+Attr+Qst, V+TV+VAbess+Qst	A+Comp+Sg+Nom+Qst, A+Comp+Sg+Nom, A+Comp+Attr, A+Comp+Attr+Qst, V+TV+VAbess
smj	N+Sg+Com+PxSg1, N+Sg+Com+PxSg2, N+Sg+Abe, N+Pl+Abe, N+Pl+Gen+PxSg1	N+Pl+Abe, N+Sg+Abe, N+Sg+Com+PxSg1, N+Sg+Com+PxSg2, N+Pl+Com+PxSg2
smn	N+Pl+Com+Qst, A+Pl+Com+Qst, A+Comp+Pl+Com+Qst, A+Superl+Pl+Com+Qst, V+PrsPrc+Qst	N+Pl+Com+Qst, N+Pl+Gen+Qst, V+Ind+Prs+Sg3+Qst, A+Pl+Com+Qst, V+Ind+Prs+ConNeg+Qst
sms	A+Superl+Sg+Abe, A+Superl+Sg+Abe+Qst/a, A+Superl+Sg+Abe+Qst/ko, V+VAbess+Qst/a, V+VAbess+Qst/ko	V+Ind+Prt+Pl1, V+VAbess+Qst/a, V+Ind+Prt+Pl1+Qst/ko, V+VAbess+Qst/ko, V+VAbess
udm	N+Sg+Ela+PxPl1, N+Sg+Ela+PxSg3, N+Sg+Ela+PxSg2, N+Sg+Ela+PxPl3+Qst, N+Sg+Ela+PxSg1	V+Ind+Prs+Pl1, V+Ind+Prs+Pl1+Qst, V+Ind+Fut+Pl1+Qst, V+Ind+Fut+Pl1, V+Imp+Pl2
vro	V+Act+Sup+Ine, V+Act+Ind+Prt+Sg2, V+Pss+Ind+Prt+Sg2, V+Pss+PrfPrc, V+Pss+PrfPrc+Sg+Nom	V+Act+Ind+Prt+Sg2, N+Pl+III, V+Pss+PrfPrc, V+Pss+PrfPrc+Sg+Nom, V+Pss+Ind+Prt+Sg2

Table 7: The top 5 morphological forms that were the most difficult to lemmatize and generate

toisimmilleän, which probably should end into *-een*. We can also observe that in many Finnish lemmas that the model does analyze correctly the forms are compounds. This leaves open the possibility that the training data has contained either the second component independently or within a comparable compound, which would have given the model some example. One lemmatization issue that can be distinguished is that the model doesn't lemmatize correctly proper names that are written with initial capital letter. These include several words, for example *Unkareinansako* 'as their Hungaries?' should be lemmatized as *Hungary*, but the model returns *nkareintaa*. What this shows is that the model struggles with uppercase characters, although those would ideally be part of the correct lemmatization result.

The Finnish model has problems in generating forms for words ending in *-lainen*, as it seems to inflect them as one would inflect the word *laine* 'wave', such as *dominikaanilaineiltasi* '≈ from

your Dominican waves' instead of *dominikaanilaisiltasi* 'from your Dominican people'. Also, other adjectives ending in *-inen* are problematic such as *keväneensä* instead of *keväisensä* 'his spring-like'. In this case, the model has not learned the typical inflectional category of adjectives ending in *-inen*. This issue has an interesting parallel with the same problem being present in the lemmatization task, described above. This shows that the problems the models encounter are to some degree parallel to one another in different tasks, and either relate to the complexity of the linguistic system, or somehow inadequately represented input.

Interestingly, the generation model has problems with the plural forms of the abessive and illative case, and often generates the singular form instead of the plural such as in *sähkömittarikesi* 'for your electricity meter' instead of *sähkömittareiksesi* 'for your electricity meters' or a completely erroneous form such as *sähkömittaritsiisi* instead

of *sähkõmittareihisi* ‘to your electricity meters’. In these erroneous cases, the model has tried to pluralize the word, for example *sähkõmittarit* is the correct plural form of electricity meters in nominative, but it is no longer correct when inflected in the illative case.

3.3.2 Komi-Zyrian

When we examine the lemmatization task, some particularities are obvious in Komi-Zyrian. For example, many of word forms with interspersed white spaces in them are not lemmatized correctly. We also see that some complex entries borrowed from Russian are challenging to lemmatize, possibly due to their rarity, for example: народно-освободительнойджыкъяснысланьджык ‘more in the direction of their people who are more national-liberational’ would correctly result in народно-освободительной, but the model predicts народнотильной. In this case the hyphen within the compound probably contributes to the rarity of the form itself. Similarly, the model is also struggling when there are words that follow orthographic conventions more typical to Russian than Komi, for example областьсаас would be correctly lemmatized as областьса, but the model predicts областььса. If this reflects the underlying code, model training like this could be very useful for locating erroneously coded transducers. The double soft sign would seem to allude to double exponence in the code. The model also has challenges with rarer orthographical conventions in Komi vocabulary. For example пипуа-кыддзаиньланьсянь ‘from the direction of my aspen and birch grove’ should be пипуа-кыддзаин ‘aspen and birch grove’, but we get пипуа-кыдзаин. These shortcomings, however, are relatively rare in the Zyrian data, and the model learns to lemmatize at high accuracy. Much more so than Finnish, which could be related to more concatenative morphology of Komi where the word boundaries can be easier to detect.

In the case of Komi-Zyrian we can observe that a large portion of wrongly recognized forms results from ambiguity that is inherent to the morphology of this language. For example, it is not possible to distinguish some of the cases, such as the inessive and illative, in all forms where they occur. As the model inevitably returns only one reading, it is clear that the evaluation accuracy cannot be perfect. This finding is consistent with

analogous ambiguity for other forms in the Skolt Sami (sms) model. There appears to be a consistency in what is incorrectly predicted in Skolt Sami. When there is a four-way ambiguity as in the *Sg Gen*, *Sg Acc Sg Nom* and *Pl Nom*, the tag *Sg Gen* is consistently predicted to be *Pl Nom*, leaving the two readings *Sg Acc* and *Sg Nom* out of the dichotomy. Komi models shows similar preferences into specific categories when there are multiple homonymous possibilities.

In the analysis above it was already briefly discussed that some categories are difficult to recognize correctly for Permic languages. Another example like this is seen in the Komi-Zyrian and Komi-Permyak (koi) future tense marking. As these languages have morphologically marked future in the third person alone, every first and second person verb in the present tense also gets a future reading, as both analyses can be seen as correct. One could also argue, however, that if some analysis is not possible to resolve at this level, some of the distinctions could be removed or merged at this level of analysis.

What comes to morphological generation of Komi, the accuracy is rather high. Some of the errors can be connected to the fact that some suffixes can occur in varying orders. For example, with input кольквиж A Sg Egr PxP11 Comp one could assume the output кольквижнымсяньджык ‘more from the direction of our yellows’, but in this case the model outputs кольквижсяньнымджык. The only difference is, however, in the order of markers for case Egr and possessive suffix PxP11. The model is actually giving a correct output, but the input doesn’t have all information about the suffix order that the model would need.

There are also instances of word generation where the correct prediction would demand actual lexicographical knowledge, which the model cannot have. For example, Komi displays with some nouns an additional stem consonant. It is not possible to predict from the surface form whether this consonant exists and what it is. So when the model is given input мек N Sg Ins, it doesn’t predict the correct мекйөн ‘with a pelt’, but offers the regular but incorrect form мекөн. This is a good example from construction where rule-formulated linguistic knowledge may be necessary for optimal analysis. It also shows that the model is capable to learn very well the regular structures of the lan-

guage and does predict them with high accuracy.

4 Conclusions

In this paper, we have presented a method for automatically extracting inflectional forms from FST transducers by composing a regular expression transducer for each word with an existing FST transducer. This way, we have been able to gather very large morphological training data for analysis, lemmatization and generation for 22 languages, 17 out of which are endangered and fighting for their survival. We have used this dataset to train neural models for each language. Because the data follows the tags and conventions used in the GiellaLT infrastructure, these neural models can be used directly side by side with the FST transducers in many of the applications that depend on them.

The results look very good for some languages while being a bit more modest for others. Analysis seems to be the hardest problem out of the three, and its training also took the longest time. Despite this, many models reached to an over 80% accuracy in the tasks. This is rather good given that the evaluation was conducted entirely on out-of-vocabulary words.

The accuracies reported in this paper are a somewhat lower than what they could be. This is due to the fact that we ran the evaluation by producing one result only for each input with the neural models and compared that input directly to the one in the test data. As we saw in our analysis, many of the inputs in the test data were ambiguous, which caused the neural model to produce an output that is correct, but not the one in the test data. However, the right way to overcome this problem would be to research how to deal with ambiguity. The neural models we trained can already now produce N best candidates for each input.

It is probable that within those N best candidates, the models actually cater for the ambiguity and produce other results that are correct as well. For instance, the Finnish word *noita*, can be an accusative singular noun meaning ‘witch’ or a partitive of *nuo* meaning ‘them’. Knowing how to maximize the number of forms the neural model produces while minimizing the number of incorrect forms is a question for another paper. Although, some methods could already be used with the models trained in this paper by in-

troducing simple modifications to how the results are predicted (Silfverberg and Tyers, 2019).

Even though we aimed for a real world scale morphological tag complexity by querying all possibilities from the FSTs, there are still a couple of morphological categories we did not tackle for practical reasons. One of them is the use of clitics. The problem with these is that they can be attached to almost any kind of word regardless of its part-of-speech and inflectional form. On top of this, multiple clitics can be added one after another. To give an idea of the scale, with clitics, Finnish has 9425 unique forms for nouns (instead of 850), 216 for adverbs (instead of 16), 14794 for adjectives (instead of 1244) and a whopping 88044 forms for verbs (instead of 6667). This means that clitics need to be solved by taking a different approach than the one we had. One could, for example, introduce some forms with different combinations of clitics here and there in the training data, in which case the question arises on how many forms need to appear with clitics in order for the model to generalize their usage.

Compounds and derivations could not be included because of how the FSTs were implemented. If you ask an FST for compounds and derivations, you will surely get them! Even in such quantities that your computer will run out of RAM and swap memory for the forms of a single word, as there is no limit to how many words can be written together to form a compound or how many times one can derive a new word from another. We people might have our cognitive limits for that, but the FSTs will not⁴. The problem of compounds is probably best to leave for a separate model to solve, as there are already methods out there for predicting word boundaries (Shao et al., 2018; Seeha et al., 2020). The compound splits by such methods could then be fed into the neural models trained in this paper. As for derivations, some of them could be included in the training data, but the question of how many forms are needed would still require further research.

⁴Take, for instance, a look at this derivational Skolt Sami word produced by the FST *Piän’njatöövölltâsttiatemesvuot’tsázvuöötövvstölškuäit’teškuättöölstölstâststööstčättömâs* for *piännai+N+Der/Dimin+Der/N2A+Der/toovvyd+Der/oollyd+Der/jed+V+Der/Caus+Der/Dimin+Der/NomAg+N+Der/Dimin+Der/N2A+Der/teqm+A+Attr+Der/vuott+N+Der/sazh+A+Err/Orth+Attr+Der/vuott+N+Der/toovvyd+Err/Orth+Der/stoollyd+V+Der/shkueqttd+Der/jed+V+Der/Caus+Der/shkueqttd+Der/oollyd+Der/stoollyd+Der/Dimin+V+Der/Dimin+Der/Dimin+V+Der/Dimin+Der/ched+Der/Caus+Der/t+A+Superl+Attr*

References

- Khalid Alnajjar, Leo Leppänen, and Hannu Toivonen. 2019. No time like the present: methods for generating colourful and factual multilingual news headlines. In *Proceedings of the 10th International Conference on Computational Creativity*. Association for Computational Creativity.
- Lene Antonsen and Chiara Argese. 2018. Using authentic texts for grammar exercises for a minority language. In *Proceedings of the 7th workshop on NLP for Computer Assisted Language Learning*, pages 1–9.
- Lene Antonsen, Trond Trosterud, and Linda Wiecheteck. 2010. Reusing grammatical resources for new languages. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*.
- Eckhard Bick and Tino Didriksen. 2015. Cg-3—beyond classical constraint grammar. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 31–39.
- Jeff Ens, Mika Härmäläinen, Jack Rueter, and Philippe Pasquier. 2019. Morphosyntactic disambiguation in an endangered language setting. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 345–349.
- Ciprian Gerstenberger, Niko Partanen, and Michael Rießler. 2017. Instant annotations in elan corpora of spoken and written komi, an endangered language of the barents sea region. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 57–66.
- Mika Härmäläinen. 2018. Poem machine—a co-creative nlg web application for poem writing. In *The 11th International Conference on Natural Language Generation Proceedings of the Conference*. The Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- José María Hoya Quecedo, Koppatz Maximilian, and Roman Yangarber. 2020. Neural disambiguation of lemma and part of speech in morphologically rich languages. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3573–3582, Marseille, France. European Language Resources Association.
- Mika Härmäläinen. 2019. UralicNLP: An NLP library for Uralic languages. *Journal of Open Source Software*, 4(37):1345.
- Mika Härmäläinen. 2021. Endangered languages are not low-resourced! In Mika Härmäläinen, Niko Partanen, and Khalid Alnajjar, editors, *Multilingual Facilitation*. Rootroo Ltd.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proc. ACL*.
- Dan Kondratyuk. 2019. Cross-lingual lemmatization and morphology tagging with two-stage multilingual BERT fine-tuning. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 12–18, Florence, Italy. Association for Computational Linguistics.
- KyungTae Lim, Niko Partanen, and Thierry Poibeau. 2018. Multilingual dependency parsing for low-resource languages: Case studies on north saami and komi-zyrian. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Kristen Lindén, Erik Axelson, Senka Drobac, Sam Hardwick, Juha Kuokkala, Jyrki Niemi, Tommi A Pirinen, and Miikka Silfverberg. 2013. HFST a system for creating NLP tools. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 53–71. Springer.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- Chaitanya Malaviya, Shijie Wu, and Ryan Cotterell. 2019. A simple joint model for improved contextual neural lemmatization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1517–1528, Minneapolis, Minnesota. Association for Computational Linguistics.
- Enrique Manjavacas, Ákos Kádár, and Mike Kestemont. 2019. Improving lemmatization of non-standard languages with joint learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1493–1503, Minneapolis, Minnesota. Association for Computational Linguistics.
- Arya D McCarthy, Christo Kirov, Matteo Grella, Amrit Nidhi, Patrick Xia, Kyle Gorman, Ekaterina Vylomova, Sabrina J Mielke, Garrett Nicolai, Miikka Silfverberg, et al. 2020. Unimorph 3.0: Universal morphology. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3922–3931.
- Sarah Moeller, Ghazaleh Kazeminejad, Andrew Cowell, and Mans Hulden. 2019. Improving low-resource morphological learning with intermediate forms from finite state transducers. In *Proceedings of the 3rd Workshop on the Use of Computational*

- Methods in the Study of Endangered Languages Volume 1 (Papers)*, pages 81–86, Honolulu. Association for Computational Linguistics.
- Christopher Moseley, editor. 2010. *Atlas of the World's Languages in Danger*, 3rd edition. UNESCO Publishing. Online version: <http://www.unesco.org/languages-atlas/>.
- Sjur Moshagen, Jack Rueter, Tommi Pirinen, Trond Trosterud, and Francis M. Tyers. 2014. Open-Source Infrastructures for Collaborative Work on Under-Resourced Languages. The LREC 2014 Workshop “CCURL 2014 - Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era”.
- Yohei Oseki, Yasutada Sudo, Hiromu Sakai, and Alec Marantz. 2019. Inverting and modeling morphological inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 170–177, Florence, Italy. Association for Computational Linguistics.
- Niko Partanen, Rogier Blokland, KyungTae Lim, Thierry Poibeau, and Michael Riebler. 2018. The first komi-zyrian universal dependencies treebanks. In *Second Workshop on Universal Dependencies (UDW 2018), November 2018, Brussels, Belgium*, pages 126–132.
- Tal Perl, Sriram Chaudhury, and Raja Giryes. 2020. Low resource sequence tagging using sentence reconstruction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2692–2698, Online. Association for Computational Linguistics.
- Tommi A Pirinen. 2019a. Building minority dependency treebanks, dictionaries and computational grammars at the same time—an experiment in karelian treebanking. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 132–136.
- Tommi A Pirinen. 2019b. Neural and rule-based finnish nlp models—expectations, experiments and experiences. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, pages 104–114.
- Tommi A Pirinen, Francis Tyers, Trond Trosterud, Ryan Johnson, Kevin Unhammer, and Tiina Puolakainen. 2017. North-sámi to finnish rule-based machine translation system. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 115–122.
- Jack Rueter and Mika Hämmäläinen. 2019. On xml-mediawiki resources, endangered languages and tei compatibility, multilingual dictionaries for endangered languages. *Rachel Edita O. ROXAS President National University (The Philippines)*, page 350.
- Jack Michael Rueter and Francis M Tyers. 2018. Towards an open-source universal-dependency treebank for erzya. In *International Workshop for Computational Linguistics of Uralic Languages*.
- Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.
- Lane Schwartz, Emily Chen, Benjamin Hunt, and Sylvia L.R. Schreiner. 2019. Bootstrapping a neural morphological analyzer for st. lawrence island yupik from a finite-state transducer. In *Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1 (Papers)*, pages 87–96, Honolulu. Association for Computational Linguistics.
- Suteera Seeha, Ivan Bilan, Liliana Mamani Sanchez, Johannes Huber, Michael Matuschek, and Hinrich Schütze. 2020. ThaiLMCut: Unsupervised pretraining for Thai word segmentation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6947–6957, Marseille, France. European Language Resources Association.
- Yan Shao, Christian Hardmeier, and Joakim Nivre. 2018. Universal word segmentation: Implementation and interpretation. *Transactions of the Association for Computational Linguistics*, 6:421–435.
- Miikka Silfverberg and Francis Tyers. 2019. Data-driven morphological analysis for uralic languages. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, pages 1–14.
- Trond Trosterud. 2004. Porting morphological analysis and disambiguation to new languages. In *SALTMIL Workshop at LREC 2004: First Steps in Language Documentation for Minority Languages*, pages 90–92. Citeseer.
- Trond Trosterud and Sjur Moshagen. 2021. Soft on errors? the correcting mechanism of a Skolt Sami speller. In Mika Hämmäläinen, Niko Partanen, and Khalid Alnajjar, editors, *Multilingual Facilitation*, pages 197–207. Rootroo Ltd.
- Francis Tyers and Mariya Sheyanova. 2017. Annotation schemes in north sámi dependency parsing. In *Proceedings of the Third Workshop on Computational Linguistics for Uralic Languages*, pages 66–75.
- Joshua Wilbur. 2018. Extracting inflectional class assignment in pite saami: Nouns, verbs and those pesky adjectives. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, pages 154–168, Helsinki, Finland. Association for Computational Linguistics.
- Xiang Yu, Ngoc Thang Vu, and Jonas Kuhn. 2020. Ensemble self-training for low-resource languages:

Grapheme-to-phoneme conversion and morphological inflection. In *Proceedings of the 17th SIG-MORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 70–78, Online. Association for Computational Linguistics.

Nasser Zalmout and Nizar Habash. 2020. Joint diacritization, lemmatization, normalization, and fine-grained morphological tagging. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8297–8307, Online. Association for Computational Linguistics.

CoDeRoomor: A new dataset for non-inflectional morphology studies of Swedish

Elena Volodina, Yousuf Ali Mohammed
University of Gothenburg / Sweden

elena.volodina@gu.se
yousuf.ali.mohammed@gu.se

Therese Lindström Tiedemann
University of Helsinki / Finland

therese.lindstromtiedemann@helsinki.fi

Abstract

The paper introduces a new resource, *CoDeRoomor*,¹ for studying the morphology of modern Swedish word formation. The approximately 16.000 lexical items in the resource have been manually segmented into word-formation morphemes, and labeled with their categories, such as prefixes, suffixes, roots, etc. Word-formation mechanisms, such as derivation and compounding have been associated with each item on the list. The article describes the selection of items for manual annotation and the principles of annotation, reports on the reliability of the manual annotation, outlines the annotation tool Legato, and presents the dataset and some first statistics. Given the "gold" nature of the resource, it is possible to use it for empirical studies as well as to develop linguistically-aware algorithms for morpheme segmentation and labeling (cf. statistical sub-word approach). The resource is freely available through Språkbanken-Text.²

1 Introduction

Linguistic complexity is a fascinating phenomenon that influences language perception, language learning and language production (cf. Housen et al., 2019; Bentz et al., 2016; Newmeyer and Preston, 2014). It has been studied at different levels and with different intentions, for example from a typological perspective (e.g. Gutierrez-Vasques and Mijangos, 2020) or from a computational perspective (e.g. Branco, 2018).

Linguistic complexity also varies between individual users of the same language, which makes

¹*CoDeRoomor* - Compounding, Derivation, Root Morphology (and more)

²<https://spraakbanken.gu.se/en/resources#refdata>

it possible to use linguistic indicators to differentiate between language typical of advanced language users as opposed to, for instance, children or beginner learners (De Clercq and Housen, 2017; Brezina and Pallotti, 2019; Pilán and Volodina, 2018).

From a second language (L2) perspective there is a need to be able to follow how the morphological complexity develops in the learner language (e.g. Pienemann and Kessler, 2012; Bonilla, 2020) for instance by following how the learner acquires more inflectional forms in the language but also by seeing how their vocabulary growth can be related to the acquisition of rules of word formation. The latter is a rather underdeveloped research area and it is that which has been our focus in developing the *CoDeRoomor* resource – we want to be able to follow how word families (cf. Bauer and Nation, 1993) grow and how awareness of word-formation mechanisms develops in language learners.

Morphology, as one of the dimensions of linguistic complexity, covers *word formation* in terms of compounding and derivational morphology as well as *inflectional morphology*, such as grammatical affixes that words take to reflect number, definiteness, gender, etc. Most publications on morphological complexity deal with studies of the inflectional dimension of morphology (e.g. Brezina and Pallotti, 2019; Forsberg and Barning, 2010), with a few rare exceptions (e.g. Bolshakova and Sapin, 2020), which is not surprising. While automatic text annotation pipelines are able to process inflectional morphology (cf. morpho-syntactic descriptors available for corpora in the Korp search interface (Borin et al., 2012, 2016)), there is a lack of corpora containing analysis of the morphemes constituting the word lemmas. This is due to the absence of gold standard resources that can be used for training automatic tools (e.g. Ketunen, 2014). This is hypothetically also the reason why we rarely find lexical resources organized

by word family principles (cf. Bauer and Nation, 1993), even though there is a clear interest in that kind of resources in connection to vocabulary testing (e.g. Sasao and Webb, 2017) and psycholinguistic and cognitive research (e.g. Amirjalili and Jabbari, 2018).³

In the currently pursued project, *Development of lexical and grammatical competences in immigrant Swedish*,⁴ funded by Riksbankens Jubileumsfond, we are looking for ways to characterize the language typical of second language (L2) learners of Swedish from different perspectives based on the analysis of two learner-specific corpora (see Section 3). Based on those corpora, we have generated a sense-based wordlist, Sen*Lex, manually segmented each item on the list into morphemes and labeled those for their morpheme categories (Section 4). The intention is to use this resource for empirical studies as well as for the development of automatic morphological segmentation and consequent morpheme classification for Swedish. We expect this type of annotation to facilitate deeper studies into lexical and morphological complexity, language acquisition patterns, associative learning mechanisms and the like. The resource can also be of interest in pedagogical studies and applications.

2 Related work

Morphemic segmentation is an important NLP task which is applied to machine translation, cognate identification, linguistic typological studies, and the like (Sennrich et al., 2015; Miestamo et al., 2008). The task of morpheme segmentation consists of the identification of morpheme boundaries within a word, and classifying them by their category. Most work has been focused on inflectional morphology and on classification of the endings by their syntactic and grammatical functions, such as gender, number, tense indicators (e.g. Cotterell et al., 2019).

Identification of word formation morphemes (roots, suffixes, prefixes) and their subsequent classification is a more complicated task, and until recently most approaches have been targeting only morpheme boundary identification using unsupervised or semi-supervised approaches, for example

³e.g. <https://www.ltu.se/research/subjects/teknisk-psykologi/nyheter/Nytt-projekt-om-barns-lasformaga-1.203355>

⁴<https://spraakbanken.gu.se/en/projects/l2profiles>

a language independent approach taken in Morfessor (Creutz and Lagus, 2007; Smit et al., 2014) or sub-word identification techniques (e.g. Gutierrez-Vasques and Mijangos, 2020).

Only recently have datasets with labeled data started to appear, and depending on their size, neural networks are used for experimentation with more complicated tasks including both morpheme segmentation and labeling of word formation morphemes (e.g. Bolshakova and Sapin, 2020; Sorokin and Kravtsova, 2018).

Morphology has not been one of the major strands of research on Swedish, neither as an L1 (native speaker language) nor as an L2 (second language learners). There also has not been a lot of interest in the development of tools and resources in relation to Swedish morphology except for Saldo morphology (Borin et al., 2013) which is used in annotation of Swedish texts and which primarily includes inflectional paradigms. Due to its language independence, Morfessor (Smit et al., 2014) offers a possibility to annotate words morphologically in any language and works relatively well on concatenative languages, including Swedish. The output consists of several suggestions for word segmentation into morpheme constituents.

In recent years interest has increased in finding ways to study different forms of complexity in connection to second language acquisition and learner corpora (Housen et al., 2019). However as Housen et al. say, morphological complexity has not been at the centre of attention. When studies have looked at morphological complexity they have also tended to focus primarily on inflectional morphology.

The resource we present in this paper is aimed at non-inflectional morphology of Swedish and can be used in a variety of NLP and linguistic tasks, including within the second language acquisition domain, and is filling a gap by offering a richly annotated dataset for morphological studies.

3 Item selection

To limit the annotation work to only the most relevant items, which in our context means items of relevance for second language learners of Swedish, we have used two source corpora:

- COCTAILL (Volodina et al., 2014), a corpus of coursebook texts that learners of Swedish

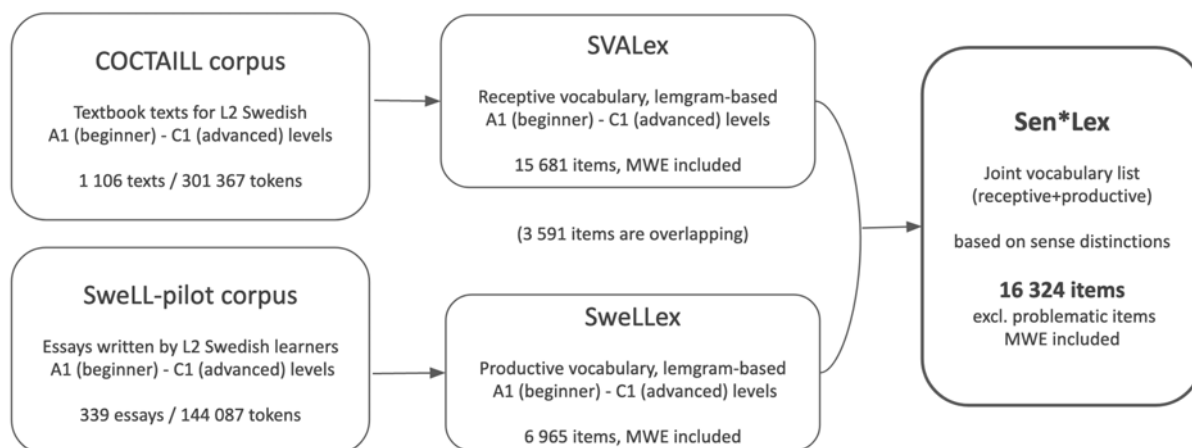


Figure 1: Selection of items for morphological annotation

as a second language (L2 Swedish) read as part of their proficiency courses, and

- SweLL-pilot (Volodina et al., 2016a), a corpus of essays written by adult learners of Swedish as a second language

Both corpora have indications of levels of proficiency according to the Common European Framework of Reference, CEFR (Council of Europe, 2001), and contain texts and essays at five out of six defined levels: A1 (beginner), A2, B1, B2, C1 (advanced).

From the two corpora, two lists of lemgrams (i.e. baseforms of the words + their corresponding parts of speech, POS) have been generated, namely:

- SVALex (François et al., 2016), consisting of L2 Swedish receptive vocabulary, and
- SweLLex (Volodina et al., 2016b), containing L2 Swedish productive vocabulary.

The approach used in the generation of the above lists has been reused by us to generate a new list based on senses (i.e. a list where each entry corresponds to a unique combination of baseform+POS+sense) once the pipeline for Swedish could assign word senses (Nieto Piña, 2019) based on Saldo senses (Borin et al., 2013). This work resulted in *Sen*Lex* (a sense-based variant of SweLLex and SVALex in one), a publication on which is currently under preparation. The non-problematic items of this latter list have been used for the morphological annotation.⁵

⁵By problematic items we mean the items that have au-

Figure 1 shows the basic information about the two source corpora and the three vocabulary lists. *Sen*Lex* includes both single-word items and multi-word expressions (MWEs), and contains word senses coming from both learner essays and course books. A certain amount of items overlap, i.e. occur in both corpora; whereas some items are homographs within the same part of speech (cf. *vara*, verb – Eng. 'be' and Eng. 'last'), but have several distinct senses. These latter items may have identical morphological analysis despite having several entries in the list, but it is also possible that they have different morphological annotation as is the case with the verb *vara*, where the root is *var-* in both lemmas but the final *-a* is seen as derivational in one sense and inflectional in the other, since *vara* (Eng. 'be'), has the imperative form *var!* and the verb *vara*, (Eng. 'last') has the stem and imperative form *vara!*, not that you are ever likely to use it in the imperative.

The *CoDeRoomor* morphological dataset that we are presenting is, thus, not all-covering for modern Swedish. However, given the nature of second language learning, the most central items should be represented in the list, therefore making it relatively comprehensive.⁶

tomatically been assigned multiple lemgrams or failed to be assigned a lemgram. These items are left for future work.

⁶We would also like to note that the set includes c. 500 triplets consisting of lemgrams which are verbs and part-of-speech-tag participle, e.g. *cykla* 'to ride a bike' + PC (participle). Since we annotate lemgrams these have then been annotated as the lemma of the verb, rather than one of the participles. We did look into annotating them as participles, but in fact each of these items can include occurrences in the data which are a combination of present participles or past participles, or even supine forms (a form etymologically re-

Word formation	Definition	Example
Abbreviation	words consisting of the initial components of a word or several words, including chemical abbreviations and some blends	AB (aktiebolag) (cf. Eng. 'ltd' = 'limited'), Au (Sw. guld, Eng. 'gold')
Compound	words formed by adding together two stems	skol+bok ('school book')
Derivation	words formed by adding a prefix or a suffix to a stem	sorg lig ('sad')
Lexicalized form	words that cannot be reduced to baseforms, e.g. MWEs	Aftonblad et (name of a tabloid), järns pikar (a swearword)
Root lexeme	words consisting of a root only or a root and an inflectional suffix	bok ('book'), adjö ('goodbye'), ande ('spirit')
Unknown	reserved for difficult or uncertain cases including most first names	alzheimers (name of a disease), kalender ('calendar')

Table 1: Taxonomy of word formation mechanisms with definitions and examples

4 Annotation principles

The aim of the morphology annotation work consisted in

- segmenting each lexical item (lemgram+POS+sense) into morphemes
- assigning a word formation description to the item according to a taxonomy (Table 1)
- categorizing each morpheme according to a taxonomy of morpheme categories (Table 2).

The items were analysed at the lemgram level and hence the work did not include annotation of inflectional forms/morphemes, with a few exceptions (see below).

For example, *oändlighet*, noun ('infinity') was

1. segmented into four morphemes
o-änd-lig-het
2. each morpheme received a label:
 - o: prefix
 - änd: root
 - lig: derivational suffix
 - het: derivational suffix
3. the word formation of the item was labeled as derivation

The taxonomy of morphemes is presented in Table 2. Most of the categories are self-explanatory, but some need to be explained.

- The category of *real root* should not be taken as representing an actual morpheme, but is used to catch cases of alternative spellings of the same root, and hence a form of allomorphy. This was done so that we could collect all words with the

related to the past participle but which only occurs in the past tense with the auxiliary verb *ha* 'to have'. We will return to these items in future work.

Morph. category	Explanation	Example
p	derivational prefix	fördjupa
r	root (orthographic)	kaotisk
rr	real root	kaos (kaotisk)
s	derivational suffix	kaotisk
f	infix*	ked j ebrev
i	inflectional suffix	i_h östas
?	unknown	ironi

Table 2: Taxonomy of morpheme categories and examples. * Swe. *fogemorfem*

same root, including alternative root spellings, into a *word family* to create a word family resource for L2 Swedish (cf. Bauer and Nation, 1993).

- The category of *inflectional suffix* was added to cover some suffixes that change in other inflectional forms in the paradigm, e.g. as the final morpheme -a in *skola* ('school', noun) since the plural is *skolor* and the compounding stem is also simply *skol*, e.g. *skol-gård* ('school yard'); and also the final morpheme -a in *läsa* since it is not part of the imperative, which in Swedish is usually seen as the verb stem *läs!* and nor is it part of the tense inflection *läser*, *läste*, *läst*. Furthermore, we needed to catch cases of lexicalized forms that are not reducible to the (otherwise existing) baseforms, e.g. *järns**pikar*** (a swear word literally meaning 'iron nails'). Yet another reason for this category was the presence of multi-word expressions, e.g. *i_det_stora_hela* ('in general'), where some of the constituent parts are always used in an inflected form whereas other parts might be possible to inflect.

- During the annotation process an additional category - question mark <?> - was introduced for dubious cases that needed further discussion, e.g.

a in *a-kassa* ('unemployment benefit fund') or the *on* in *ironi*, *ironisk* ('ironic, ironical'). In most cases later comparisons helped resolve these issues and enabled the classification of the morpheme into one of the main morpheme categories.

The taxonomy of word formation mechanisms follows from Table 1, and is based on SAG (Teleman et al., 1999) and Haspelmath (2002). Where a word was a derivation based on a compound (e.g. *all-var-lig*, 'serious') or a compound which consisted partly of a derivation (e.g. *å-bäk-e*, 'monstrosity'), only word formation mechanism that gave us the final word was annotated, i.e. *all-var-lig* was annotated as a derivation and *å-bäk-e* as a compound. Detailed description of our annotation principles is available in our guidelines.⁷

To prepare a reliable resource for analysis of Swedish morphology, two authoritative resources have been used for major guidance in our annotation work: *the Swedish Academy Grammar*, SAG (Teleman et al., 1999) and two contemporary lexicons from the Swedish Academy: *the Contemporary Dictionary of the Swedish Academy*, *Svensk ordbok*, SO and *The Swedish Academy Glossary*, *Svenska akademiens ordlista*, SAOL (Sköldberg et al., 2019), both available through <https://svenska.se/>. To get access to the information in the lexicons, the Swedish Academy further allowed us to match our list of items against the SO/SAOL database, download the aspects of interest and integrate them into our annotation tool where annotators could consult them or copy to work further based on that.⁸

Each item in the SO/SAOL database contains division markers within the word, indicating where two morphemes meet (see Figure 2). Dots and vertical lines are used as notations, where the vertical line has a higher priority and is seen as a major word boundary. However, no information is provided about exactly what each morpheme stands for, e.g. whether it is a derivational suffix,

SAOL	SO
tryckår: 2015 pro-centu-ell [-el] adjektiv -t -g • räknad el. uttryckt i procent Visa mer	tryckår: 2009 procentuell adjektiv -t ORDLED: pro-centu-ell • som räknas i procent Visa mer
tryckår: 2015 kär-leks af-fär substantiv -er -er • till affär 4	tryckår: 2009 kärleksaffär substantiv -er -er ORDLED: kär-leks af-fär-en

Figure 2: SO-SAOL analysis

an inflection or a root. They also do not provide marking of the compounding / derivational infix (Swe. *fogemorfem*), since their notation has the primary goal to indicate to the user where a word can be hyphenated, and infixes are always then attached to the stem.

5 Annotation workflow and visualization

A team of three highly qualified annotators performed the annotation under the supervision of a project researcher. During the first month the three annotators went through a training period where they worked in parallel with the project researcher and annotated 100 new items per week plus reannotated items from previous weeks when need be. Based on the parallel items, comparisons were run on both the morphological analysis and the word formation assignment of each item. The guidelines were refined to take care of any remaining unclarities or disagreements.

The 400 items that were annotated by the 4 members of the morphology group during the training period have been used for calculating Inter-Annotator Agreement (IAA) which we report using Krippendorff's Alpha in Table 3. As can be seen from the Table, the agreement was consistently high during all training steps, with segmentation being the most agreed upon annotation type (0.93) and labeling the one with most disagreements (0.86). However, the agreement is considered to be acceptable with values over 0.75, and very high with values over 0.9, which makes us believe that the annotation of *CoDeRoomor* is very reliable and of high quality.

After annotating 400 items in parallel, the rest of the items were divided between the 3 annotators with weekly meetings to monitor progress and discuss problematic cases. Before each meeting the project researcher got a morpheme-based

⁷<https://docs.google.com/document/d/1G5PEfeDEKq4dAZaupj6FmUUWBGiegigagzXgTA3cDSY/>

⁸We initially discussed an opportunity to use automatic pre-processing for detection of morpheme boundaries, e.g. using SWETWOL tool (Karlsson, 1992) or Morfessor (Creutz and Lagus, 2007), but instead opted for expert morpheme boundary indication performed by trained lexicographers and available through the SAOL/SO, as described in this subsection.

Annotation type	1-100	-200	-300	-400
Segmentation	0.87	0.87	0.89	0.93
Labeling	0.86	0.86	0.89	0.86
Segmentation+Labeling	0.85	0.85	0.87	0.88
Word formation	0.89	0.89	0.94	0.91

Table 3: IAA measure using Krippendorff’s Alpha, reported for each 100-word portion.

comparison where disagreements and partial disagreements were identified, and these could then be checked by the project researcher and discussed as needed at the meeting. The team came up with a solution and amended the guidelines to ensure systematic annotation in the future. After each meeting the annotators were expected to correct any items that had been picked up in the comparison to adhere to the agreed or revised principles. The guidelines have been a living document all through the process. An article with a more detailed description of the linguistic principles of segmentation and labeling is under preparation (Lindström Tiedemann et al., In Prep.). Once the annotation was completed the project researcher once again checked disagreements, partial disagreements and also searched through the data consistently for certain strings to find possible inconsistencies. Based on this some further corrections were done according to the guidelines, e.g. if a suffix had been annotated as an inflectional suffix but should be a derivational suffix according to the guidelines this was corrected.

To ensure consistency of the annotation work, a tool for lexicographic annotation *Legato* (Alfter et al., 2019) was implemented within the framework of the project, see Appendix A. The tool requires annotators to log in to save their annotations. The functionality of the tool allows the annotator to see

- the current item as a lemgram, the lemgram part of speech, the part of speech tag and its first level of occurrence in the source corpora
- sense descriptor from the Saldo lexicon
- examples from the corpora
- two fields where previous annotation for the annotator appears when available
- two fields with annotations from the Swedish Academy lexicons (SO and SAOL)
- a text area for entering "Current values" for the analysis

In addition, the tool offers possibilities to open guidelines, check a list of previously "skipped items" or click on supportive links (among others, COCTAILL corpus hits for the current item and SAOL/SO hits). To navigate between the items, it is possible to "jump" to another item at a certain numeric index, search for some specific items or filter items.

Furthermore, the tool also allows each annotator to download their own annotated words with time stamps for inspection of the results. The project researchers can, in addition, download the annotations from all annotators, to generate several types of comparisons and statistics, and download a full set of annotated words.

6 *CoDeRoomor* dataset description

The *CoDeRoomor* dataset (version 1.0) contains 16 230 analyzed lemgrams⁹ representing 4 429 unique roots, 259 unique derivational suffixes, 155 unique prefixes and 12 unique binding morphemes (infixes), see Table 4. Table 4 shows statistics over all morphemes in the dataset with some examples, number of times these morphemes appear in the lexemes in the Sen*Lex list, number of times they are used in the running tokens in the COCTAILL corpus (coursebooks) and in the SweLL-pilot corpus¹⁰ (essays).

The five most frequent root morphemes in the Sen*Lex items on the *CoDeRoomor* are:

- *ut* (313 words in the "family", each containing that root), e.g. *utbildning*, ('education')
- *i* (272 words), e.g. *i* (preposition), ('in')
- *för* (228 words), e.g. *överföra*, ('transfer')
- *upp* (225 words), e.g. *kolla upp*, *uppdrag*, ('check up', 'assignment')
- *till* (189 words), e.g. *tillbaka*, ('back')

If we instead look at the five most frequent root morphemes in the corpora, the most common in Coctail are *ha* (13 933 words), *var* (13 597

⁹We started with 16 324 triplets (lemgram + POS + sense), but we had to invalidate some lemgrams which were incorrectly lemmatized and not found in the data when doublechecking. We found these items since they were unexpected in learner data and were therefore doublechecked in the corpora by the project researcher supervising the annotation.

¹⁰The calculations were performed on a new version of SweLL-pilot, from 2020, which contains an extended collection of essays compared to Volodina et al. (2016a), namely 490 essays and 156 988 tokens (as compared to 339 essays and 144 087 tokens in the 1st version)

Morpheme category	Unique count	Sen*Lex	COCTAILL	SweLL-pilot	Examples
root	4429	23 987	471 056	142 381	matbord, kärleksaffär, sagolik
suffix	259	10 062	91 646	28 638	marknad, kostsam, militär
prefix	155	2 183	19 828	5 489	konsonant, nyrenoverad
infix	12	1 089	3 441	1 641	kännedom, kvinnorörelse
inflection	32	3 067	88 641	28 810	saker_och_ting, Medelhavet, läsa

Table 4: Statistics per morpheme type in the three resources

words), och (13 154 words), gå (13 046 words) and kunn (12 528 words). In the SweLL-pilot they are kunn (6 962 words), att (5 007 words), och (4 737 words), var (4 726 words) and jag (4 690 words).

Examples of other frequent root morphemes are

- liv with 73 family members, e.g. affärsliv, livmoder, leva.livet, ('business life', 'uterus', 'live.the.life'); and
- sam with 53 family members, e.g. samtal, samhällelig, sambo, ('conversation', 'societal', 'partner').

On inspection we can see that these family groupings need to be refined to be separated further into proper "families", so that words containing unrelated homographic roots do not accidentally end up in the same family. To give one example, the sam-family at the moment contains both samisk ('Sami', adj.) and samhälle ('society'), which should be separated into two different families since a morpheme is the smallest meaningful unit in language and therefore each root should have only one meaning and homographs should be separated.

Taking a look at the most frequent derivational morphemes (prefixes and suffixes), we can see that in the annotated wordlist

- the most common prefixes in the wordlist are för- (380 words), be- (299 words), o- (256 words), re- (112 words), pro- (85 words) as in förälder, besök, odjur, reagera, problem ('parent', 'visit', 'beast', 'react', 'conference', 'problem')
- and the most common derivational suffixes in the wordlist are -a (1 894 words), -er (640 words), -ning (443 words), -ig (433 words), -ar (378 words) as in idrotta, aktivera, utbildning, duktig, ägare ('do sports', 'activate', 'education', 'smart', 'owner').

There are several prefixes that only occur

once in the dataset (wordlist and corpora), e.g. abs-, fysio-, ko- as in abstrakt, fysiologisk, koefficient ('abstract', 'physiological', 'coefficient'). In cases such as koefficient it would be good to consider comparison to allomorphs such as kon, but this currently needs to be done manually. Some of the least common suffixes are -ej, -enn as in pastej, persienn ('paté', 'Venetian blind') which only occur once in the dataset (wordlist and corpora). In the dataset it is also possible to access the frequency in relation to the number of occurrences in the L2 corpora we work with.

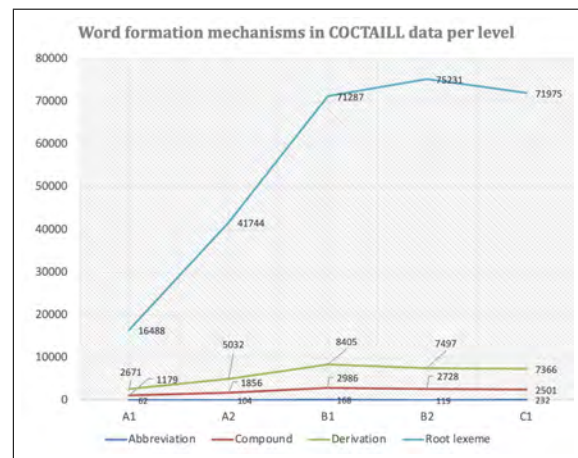


Figure 3: Statistics (raw count) over word formation mechanisms in the course book data

From the initial exploration of the word formation mechanisms in the two source corpora, we can see that *root lexemes* clearly dominate (Figures 3 and 4), followed by *derivation* and *compounding*. *Abbreviation* is hardly represented, nor are *lexicalized forms* that we haven't even included into the graphs. The hypothetical reason for the overrepresentation of *root lexemes* can be the fact that most frequent words in the language, namely prepositions, particles and conjunctions, are root lexemes and therefore add to the running statis-

Lemgram	Sense	POS	Analysis	Segment.	Pattern	RealRoot	WordForm	CEFR
adekvat..av.1	adekvat..1	JJ	p:ad r:ekv s:at	ad-ekv-at	p:r:s		derivation	C1
adla..vb.1	adla..1	PC	r:adl s:a	adl-a	r:s	rr:adel	derivation	B2
adel..nn.1	adel..1	NN	r:adel	adel	r		root_lexeme	B1
adelsman..nn.1	adelsman..1	NN	r:adel f:s r:man	adel-s-man	r:f:r		compound	B1
adjektiv..nn.1	adjektiv..1	NN	p:ad r:jekt s:iv	ad-jekt-iv	p:r:s		derivation	A2
adjö..in.1	adjö..1	IN	r:adjö	adjö	r		root_lexeme	A2

Table 5: *CoDeRooMor* dataset by lemgram, an excerpt

Morpheme	Identifier	Category	Frequency	Examples
a	s	suffix	1 605	leverera, lugna_sig, meritera, narkotika, pumpa, rasa
er	s	suffix	577	abdikera, intrigera, politiker, kritiker, motivera, tekniker
tid	r	root	128	arbetstid, nutid, skoltid, livstid, dåtid, deltid
ny	r	root	46	nyinköpt, nykockt, nykomling, nyligen, nymodighet, Nynäshamn
o	p	prefix	240	olaglig, olämplig, olik, olika, olikhet, oljud
re	p	prefix	105	reaktionstid, rebell, rebellisk, recensent, recensera, recension
s	f	infix*	803	fredstid, landsfader, riksbank, tvångsgift, landsdel, riksdag
o	f	infix*	76	vilopaus, sagobok, sannolik, sociolog, vilorum, typografi

Table 6: *CoDeRooMor* dataset by morpheme with examples, an excerpt. *Swe. *fogemorfem*

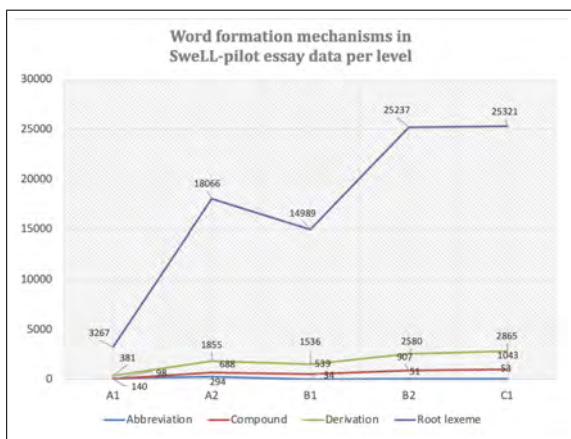


Figure 4: Statistics (raw count) over word formation mechanisms in the learner essay data

tics. In addition some words which could also be seen as derivations are currently seen as root lexemes since the final suffix falls in other inflectional forms and hence they are counted as root lexemes, e.g. *resa* 'to travel', cf. *resa* 'journey', since the rule was that annotators should usually select a word formation which fit with the first part of the annotation (segmentation and morpheme categorization).

Using *CoDeRooMor*, it is possible to trace the morphemic complexity of the words at different stages of language development. From Figure 5 we can see that the morphemic word structure is getting more complex as proficiency develops, with the average number of morphemes per new word (based on words which first occur at that CEFR-level in our data) growing from 1.79 at the

Item	A1	A2	B1	B2	C1	Total
word	1369	2689	4518	4440	3211	16230
morpheme	2457	5727	11318	11732	9292	40534
morpheme/word	1.79	2.13	2.51	2.64	2.89	2.50

Figure 5: Statistics over morpheme per word at different levels of proficiency

beginner level till 2.89 at the advanced level.

The dataset can be downloaded as an excel file or as a file with comma separated values (csv file format). The information can be organized in several ways:

- with lemgrams as the main lookup items (see Table 5). The associated information per lemgram consists of:
 - lemgram
 - sense indicator (Saldo-based)
 - part of speech
 - analysis by morpheme
 - real root (if applicable)
 - word segmentation boundaries
 - word morpheme patterns
 - word formation mechanism
 - the CEFR level (level of first occurrence)
 - frequency information from COC-TAILL, by level and in total (if applicable)
 - frequency information from SweLL-pilot, by level and in total (if applicable)

2. with morphemes as the main lookup item (see Table 6 for an example). The associated information consists of:

- morpheme, e.g. `abs`
- identifier, e.g. `p`
- category, e.g. `prefix`
- number of unique words containing that morpheme in the Sen*Lex list
- list of words containing this morpheme-category (building a "morpheme" family)
- frequency in Sen*Lex by level and in total, if applicable (several columns)
- frequency in COCTAILL by level and in total, if applicable (several columns)
- frequency in SweLL-pilot by level (several columns)

The Legato annotation tool can compile some statistics and tables for overviews and visualization, which currently is only available for project researchers. In the future, we plan to make these functions open to all users, together with making this dataset available not only for download, but also for browsing (cf. English Vocabulary Profile, Capel, 2010).

The *CoDeRoomor* dataset can be freely downloaded from Språkbanken-Text.¹¹

7 Future work

The *CoDeRoomor* resource offers promising possibilities for several types of research. Research questions with Linguistics and Second Language Acquisition domain are described in detail in Lindström Tiedemann et al. (In Prep.) and are mentioned briefly in the introduction to this article. With regards to pedagogical and applied research prospects, we are currently exploring how the items can best be linked together and presented to the public as a word family resource for use both in research and in teaching. The plan is that since Swedish uses both derivation and compounding frequently the resource will show all words which have a common root as a family and there will be information about how this relates to CEFR levels based on the corpora that we mentioned above. The dataset can be effectively used for Intelligent Computer-Assisted Language Learning research,

¹¹<https://spraakbanken.gu.se/en/resources#refdata>

for example for exercise generation or text complexity analysis.

To visualize the resource and support research into the non-inflectional morphology, we are working on a user interface for Swedish similar to the English Vocabulary Profile (EVP)¹² and Pearson GSE Teacher Toolkit.¹³ The interface has a working title *Swedish L2 Profile* (SweL2P) and is integrated into the Lärka platform¹⁴ (Alfter et al., 2018), at Språkbanken Text (Gothenburg, Sweden). The GUI will provide possibilities to search, filter, browse and download various L2 Swedish datasets (lexical, morphological, grammar, including *CoDeRoomor*) generated as an output of the project.

We are currently also experimenting with automatic morpheme segmentation based on the *CoDeRoomor* dataset which is showing promising results and we hope that this might result in a new functionality in the Sparv pipeline (Borin et al., 2016) allowing automatic segmentation and labeling of morpheme categories for Swedish.

The ultimate aim is to analyze learner language in a more nuanced way, where analysis of word formation morphemes could help us to look deeper into lexical and morphemic complexity and to understand language acquisition and processing better. Type token ratio (TTR) has been often used as a way to measure lexical diversity, i.e. how varied the vocabulary in a text is (see e.g. McKee et al., 2000). However recently TTR has also been used as a means of studying morphological complexity (Gutierrez-Vasques and Mijangos, 2020; Ketunen, 2014). Gutierrez et al. also explore the possibility of studying morphological complexity through entropy and CRF in relation to typological comparisons of languages. Our intention is to apply similar techniques for analysis of learner language.

Acknowledgments

This work has been supported by a grant from the Swedish Riksbankens Jubileumsfond (Development of lexical and grammatical competences in immigrant Swedish, project P17-0716:1). We acknowledge the project assistants – Beatrice Silén, Stellan Petersson and Maisa Lauriala – for their thorough annotation work; and David Alfter for

¹²<http://www.englishprofile.org/wordlists>

¹³<https://www.english.com/gse/teacher-toolkit/user/lo>

¹⁴<https://spraakbanken.gu.se/larkalabb/svlp>

the initial implementation of the Legato tool and the generation of the Sen*Lex list. We thank the Swedish Academy and the SAOL/SO group at the University of Gothenburg for sharing parts of their valuable datasets with us.

References

- David Alfter, Lars Borin, Ildikó Pilán, Therese Lindström Tiedemann, and Elena Volodina. 2018. From language learning platform to infrastructure for research on language learning. In *CLARIN Annual Conference 2018*.
- David Alfter, Therese Lindström Tiedemann, and Elena Volodina. 2019. Legato: A flexible lexicographic annotation tool. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 382–388.
- Forough Amirjalili and Ali Akbar Jabbari. 2018. The impact of morphological instruction on morphological awareness and reading comprehension of efl learners. *Cogent Education*, 5(1):1523975.
- Laurie Bauer and Paul Nation. 1993. Word families. *International journal of Lexicography*, 6(4):253–279.
- Christian Bentz, Tatjana Soldatova, Alexander Kopenig, and Tanja Samardžić. 2016. A comparison between morphological complexity measures: typological data vs. language corpora. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CLALC). Osaka, Japan, December 11-17 2016*.
- Elena Bolshakova and Alexander Sapin. 2020. An experimental study of neural morpheme segmentation models for russian word forms.
- Carrie Bonilla. 2020. Processability theory and corpora. *The Routledge Handbook of Second Language Acquisition and Corpora*, page 201.
- Lars Borin, Markus Forsberg, Martin Hammarstedt, Dan Rosén, Roland Schäfer, and Anne Schumacher. 2016. Sparv: Språkbanken’s corpus annotation pipeline infrastructure. In *The Sixth Swedish Language Technology Conference (SLTC), Umeå University*, pages 17–18.
- Lars Borin, Markus Forsberg, and Lennart Lönngrén. 2013. SALDO: a touch of yin to WordNet’s yang. *Language resources and evaluation*, 47(4):1191–1211.
- Lars Borin, Markus Forsberg, and Johan Roxendal. 2012. Korp-the corpus infrastructure of språkbanken. In *LREC*, pages 474–478.
- António Branco. 2018. Computational complexity of natural languages: A reasoned overview. In *Proceedings of the Workshop on Linguistic Complexity and Natural Language Processing*, pages 10–19, Santa Fe, New-Mexico. Association for Computational Linguistics.
- Vaclav Brezina and Gabriele Pallotti. 2019. Morphological complexity in written l2 texts. *Second language research*, 35(1):99–119.
- Annette Capel. 2010. A1–B2 vocabulary: insights and issues arising from the English Profile Wordlists project. *English Profile Journal*, 1.
- Ryan Cotterell, Christo Kirov, Mans Hulden, and Jason Eisner. 2019. On the complexity and typology of inflectional morphological systems. *Transactions of the Association for Computational Linguistics*, 7:327–342.
- Council of Europe. 2001. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Press Syndicate of the University of Cambridge.
- Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1):1–34.
- Bastien De Clercq and Alex Housen. 2017. A cross-linguistic perspective on syntactic complexity in l2 development: Syntactic elaboration and diversity. *The Modern Language Journal*, 101(2):315–334.
- Fanny Forsberg and Inge Bartning. 2010. Can linguistic features discriminate between the communicative CEFR-levels?: A pilot study of written L2 French.
- Thomas François, Elena Volodina, Ildikó Pilán, and Anaïs Tack. 2016. SVALex: a CEFR-graded Lexical Resource for Swedish Foreign and Second Language Learners. In *LREC*.
- Ximena Gutierrez-Vasques and Victor Mijangos. 2020. Productivity and predictability for measuring morphological complexity. *Entropy*, 22(1):48.
- Martin Haspelmath. 2002. *Understanding morphology*. Routledge.
- Alex Housen, Bastien De Clercq, Folkert Kuiken, and Ineke Vedder. 2019. Multiple approaches to complexity in second language research. *Second language research*, 35(1):3–21.
- Fred Karlsson. 1992. SWETWOL: A comprehensive morphological analyser for Swedish. *Nordic Journal of Linguistics*, 15(1):1–45.
- Kimmo Kettunen. 2014. Can type-token ratio be used to show morphological complexity of languages? *Journal of Quantitative Linguistics*, 21(3):223–245.
- Therese Lindström Tiedemann, Beatrice Silén, Maisa Lauriala, Stellan Petersson, Yousuf Ali Mohammed, and Elena Volodina. In Prep. Swedish morphology and Second language acquisition.

- Gerard McKee, David Malvern, and Brian Richards. 2000. Measuring vocabulary diversity using dedicated software. *Literary and linguistic computing*, 15(3):323–338.
- Matti Miestamo et al. 2008. Grammatical complexity in a cross-linguistic perspective. *Language complexity: Typology, contact, change*, 23:41.
- Frederick J Newmeyer and Laurel B Preston. 2014. *Measuring grammatical complexity*. Oxford University Press, USA.
- Luis Nieto Piña. 2019. *Splitting rocks: Learning word sense representations from corpora and lexica*. Doctoral Thesis, University of Gothenburg.
- Manfred Pienemann and Jörg-U Kessler. 2012. Processability theory. *The Routledge handbook of second language acquisition*, pages 228–247.
- Ildikó Pilán and Elena Volodina. 2018. Investigating the importance of linguistic complexity features across different datasets related to language learning. In *Proceedings of the Workshop on Linguistic Complexity and Natural Language Processing*, pages 49–58.
- Yosuke Sasao and Stuart Webb. 2017. The word part levels test. *Language Teaching Research*, 21(1):12–30.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- Emma Sköldberg, Louise Holmer, Elena Volodina, and Ildikó Pilán. 2019. State-of-the-art on monolingual lexicography for Sweden. *Slovenščina 2.0: empirical, applied and interdisciplinary research*, 7(1):13–24.
- Peter Smit, Sami Virpioja, Stig-Arne Grönroos, Mikko Kurimo, et al. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *The 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Gothenburg, Sweden, April 26-30, 2014*. Aalto University.
- Alexey Sorokin and Anastasia Kravtsova. 2018. Deep convolutional networks for supervised morpheme segmentation of russian language. In *Conference on Artificial Intelligence and Natural Language*, pages 3–10. Springer.
- Ulf Teleman, Staffan Hellberg, and Erik Andersson. 1999. *Svenska akademiens grammatik*. Svenska akademien.
- Elena Volodina, Ildikó Pilán, Stian Rødven Eide, and Hannes Heidarsson. 2014. You get what you annotate: a pedagogically annotated corpus of coursebooks for Swedish as a Second Language. In *Proceedings of the third workshop on NLP for computer-assisted language learning at SLTC 2014, Uppsala University*, 107. Linköping University Electronic Press.
- Elena Volodina, Ildikó Pilán, Ingegerd Enström, Lorena Llozhi, Peter Lundkvist, Gunlög Sundberg, and Monica Sandell. 2016a. Swell on the rise: Swedish learner language corpus for European reference level studies. *LREC 2016*.
- Elena Volodina, Ildikó Pilán, Lorena Llozhi, Baptiste Degryse, and Thomas François. 2016b. SweLLex: second language learners’ productive vocabulary. In *Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition at SLTC, Umeå*, 130, pages 76–84. Linköping University Electronic Press.

APPENDICES

APPENDIX A

Guidelines Skipped items ⁷⁹ Search Filter External links

Quick jump to:
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Å Ä Ö

Jump to: Jump

1 - 16314

Current task: **MORPHOLOGY** Progress: 134/16314

SALDO lemmgram	Saldo POS	Part-of-Speech	CEFR level
alkoholism..nn.1	noun (nn)	Noun (NN)	B2

Saldo sense: **alkoholism..1**
Saldo primary descriptor: **missbruk..1**
Saldo secondary descriptor: **alkohol..1**

Examples:
De flesta svenskar var fattiga , och sjukdomar och ** alkoholism ** gjorde livet svårt för många .
** Alkoholismen ** var utbredd och dryckenskap och slagsmål hängde ofta ihop .

Previous morphology1:

Previous morphology2:

SO analysis: alko·hol·ism·en

SAOL analysis: alko·hol·ism

Current values:
r:alko
s:hol
s:ism

Figure 6: Annotation tool, morphology annotation task

Chunking Historical German

Katrin Ortmann

Department of Linguistics

Fakultät für Philologie

Ruhr-Universität Bochum

ortmann@linguistics.rub.de

Abstract

Quantitative studies of historical syntax require large amounts of syntactically annotated data, which are rarely available. The application of NLP methods could reduce manual annotation effort, provided that they achieve sufficient levels of accuracy. The present study investigates the automatic identification of chunks in historical German texts. Because no training data exists for this task, chunks are extracted from modern and historical constituency treebanks and used to train a CRF-based neural sequence labeling tool. The evaluation shows that the neural chunker outperforms an unlexicalized baseline and achieves overall F-scores between 90% and 94% for different historical data sets when POS tags are used as feature. The conducted experiments demonstrate the usefulness of including historical training data while also highlighting the importance of reducing boundary errors to improve annotation precision.

1 Introduction

The analysis of linguistic phenomena in historical language requires large amounts of annotated data. For example, to study the development of syntactic phenomena like object order or extraposition in German, syntactically annotated texts from all relevant time periods are needed. To date, however, only very few historical corpora provide annotations beyond the morpho-syntactic level, thus limiting syntactic research to qualitative studies on small data sets. Using NLP methods for the automatic creation of relevant annotations could support the annotation process and reduce the necessary manual effort for quantitative studies. But the application of standard tools to historical data

faces a variety of challenges, as there is less or no training data, the data is less standardized, etc.

The present study investigates the automatic recognition of chunks in historical German. Section 2 gives a short introduction to the chunking task and explains peculiarities about chunking German concerning complex pre-nominal modification. Section 3 presents previous approaches to automatic chunking, which have not yet been applied to historical data, likely because no manually annotated data is available. In this study, to address the lack of chunked historical data, chunks are extracted from modern and historical constituency treebanks. Section 4 describes the training data as well as the additional test data sets before Section 5 introduces the selected methods for automatic chunking: a regular expression-based baseline and a neural CRF chunker. Finally, Section 6 details the evaluation process and presents the results, followed by a conclusion in Section 7.

2 Chunking (German)

Chunking is also referred to as partial or shallow parsing. The concept of chunks was introduced by Abney (1991), who defines them as non-recursive phrases from a sentence's parse tree ending with the head of the phrase. According to this definition, a chunk may contain chunks of other types but not of the same type, and post-nominal modifiers start a new chunk. Example (1) shows the annotation of an English sentence following Abney's chunk definition:

- (1) [S [NP The woman] [PP in [NP the lab coat]]
[VP thought]] [S [NP you] [VP had bought]
[NP an [ADJP expensive] book]].

(Kübler et al., 2010, p. 147)

The CoNLL-2000 shared task on chunking (Sang and Buchholz, 2000), which is still widely used as a benchmark, has popularized a more restricted definition of chunks and only allows

for non-recursive, non-overlapping chunks, i.e. a word belongs to a maximum of one chunk while keeping the restriction that a chunk ends at the head token. When applied to sentence (1), this results in the annotation in example (2).

- (2) [NP The woman] [PP in] [NP the lab coat] [VP thought] [NP you] [VP had bought] [NP an expensive book].

Defining chunks this way makes them suitable for the automatic annotation with sequence labeling methods and is especially useful for tasks that do not require a complete syntactic analysis but profit from an easy and fast annotation, e.g. agreement checking in word processors (Fliedner, 2002; Mahlow and Piotrowski, 2010). Furthermore, it may serve as a basis for deeper syntactic analyses (cf. Van Asch and Daelemans, 2009; Daum et al., 2003; Osenova and Simov, 2003) and thus could build the foundation for the automatic syntactic annotation of historical data.

However, applying the standard definition of chunks is problematic when chunking German because of possibly complex pre-nominal modification. The phrase in example (3) violates Abney’s chunk definition due to the embedded noun chunk and, when annotated according to the CoNLL-style definition, it would contain an article *der* that is separated from its noun chunk as in example (4).

- (3) [NC der [NC seinen Sohn] liebende Vater]
the his son loving father
‘the father who loves his son’

(Kübler et al., 2010, p. 148)

- (4) **der** [NC seinen Sohn] [NC liebende Vater]

While in some German corpora, these stranded tokens are left unannotated, e.g. DeReKo (Dipper et al., 2002), Kübler et al. (2010) introduce a special category for stranded material, marked with an initial ‘s’, e.g. *sNC* for a stranded noun chunk. They also suggest including the head noun chunk in the prepositional chunk while leaving post-nominal modifiers separate. In the following, their approach is adopted for chunking German.

Of the eleven original chunk types from the CoNLL-2000 shared task, four main types are considered in this study: noun chunks (NC), prepositional chunks (PC), adjective chunks (AC), and adverb chunks (ADVC), and, in addition, stranded noun (*sNC*) and prepositional chunks (*sPC*). Example (5) shows the annotation of a sentence from

an 1871 newspaper taken from one of the historical data sets in this study. For better readability, the relation of stranded articles to their respective noun chunks is indicated by subscripts.

- (5) [*sNC*₁ die] [*sNC*₂ den] [PC an Deutschland] [NC₂ abgetretenen Landesteilen] [NC₁ angehörenden Kriegsgefangenen] [...] werden [ADVC sofort] [PC in Freiheit] gesetzt;
the the to Germany transferred territories belonging prisoners of war will be immediately to freedom set
‘Prisoners of war belonging to the territories transferred to Germany will be released immediately’

Allgemeine Zeitung, no. 72, 1871
(DTA; BBAW, 2021)

3 Related Work

Since chunking can be understood as both a shallow parsing and a sequence labeling task, depending on the chunk definition, there have been many different approaches to the automatic identification of chunks. For non-recursive Abney-style chunking, Abney (1991) uses finite-state cascades, yet similar techniques have also been applied to CoNLL-style chunking. Müller (2005) gives an overview of chunking studies on German, many of which use finite state-based methods, but also other parsing approaches. For his FSA-based chunker, he reports an overall F_1 -score of 96%.

For non-recursive, non-overlapping CoNLL-style chunking, there have been experiments with different classification and sequence labeling methods, including the application of taggers (e.g. Osborne, 2000; Molina and Pla, 2002; Shen and Sarkar, 2005) with F_1 -scores between 92% and 94% as well as machine learning, e.g. with Conditional Random Fields yielding F_1 -scores of 93% to 94% (cf. Sun et al., 2008; Roth and Clematide, 2014). More recently, there have also been experiments with neural sequence labeling using bi-directional LSTMs (Akhundov et al., 2018; Zhai et al., 2017), RNNs (Peters et al., 2017), and neural CRFs (Huang et al., 2015; Yang and Zhang, 2018) with F_1 -scores of about 95%.

As chunks of a given type can only contain certain part-of-speech sequences, most of the studies use POS tags as features. However, lexicalization of models can also improve chunking results (cf. Shen and Sarkar, 2005; Indig, 2017) and current

contextual word representations already seem to have some awareness of shallow syntactic structures like chunks (Swayamdipta et al., 2019). In general, van den Bosch and Buchholz (2002) find that POS tags are most relevant if the training data is small, while words become more helpful with increasing amounts of data, and a combination of both features yields the best results.

For evaluation, most studies still use the data set from the CoNLL-2000 shared task (Sang and Buchholz, 2000), i.e. WSJ data from the Penn Treebank, and written news data also serves as the evaluation basis for most studies on German. However, when Pinto et al. (2016) compare tools on English CoNLL-2000 data with their performance on Twitter data, they find that for standard toolkits, F_1 -scores decrease by 17 to 38 percentage points to 45%–54% on social media text. A similar drop in performance might also occur for other non-standard data like historical language and would underline the importance of methods and models that are specifically tailored to a particular language variety.

But to date, there has only been a small number of studies on the automatic syntactic analysis of historical German, all of which have to deal with a lack of syntactically annotated historical data. In the absence of a gold standard, some studies develop rule-based approaches, e.g. Chiarcos et al. (2018) for topological field identification in Middle High German. But without the possibility for evaluation, the accuracy of such systems remains unclear. Other studies try to compensate for the lack of historical data by falling back on modern German. Petran (2012) approximates historical language by removing punctuation and capitalization from a modern German news corpus. Using CRFs, he tries to identify segments of increasing length, chunks, clauses, and sentences, in this artificial data set and concludes that smaller units are easier to identify. For chunking, he reports an F_1 -score of 93.3%, but since capitalization and punctuation are not the only differences between modern and historical German, it is unclear how well these results generalize to real historical data. Nevertheless, the exploitation of modern data can be conducive for automatically annotating historical language by reducing the need for large annotated historical data sets. As a previous study has shown, models trained on modern newspaper text can successfully be transferred to historical

German with F_1 -scores $>92\%$ when POS tags are used as input unless the historical language structures differ too much from modern German (Ortmann, 2020).

4 Data

As already mentioned, most German corpora and especially historical corpora do not offer a manual chunk annotation that could be used for training and evaluating automatic models. However, Kübler et al. (2010) notice that chunks can be extracted directly from constituency trees by converting the lowest phrasal projections with lexical content to chunks. Using this method, they automatically transform the constituency annotations from the TüBa-D/Z treebank (Telljohann et al., 2017) into chunks. The resulting corpus¹ comprises 3,816 newspaper articles with more than 100k sentences and almost 2M tokens. In total, it contains over 743k instances of the six chunk types considered in the present study.

Since the extracted chunks might be influenced by the structure of the constituency trees and, hence, may differ between treebanks with different syntactic annotation schemes, a second German treebank is included in the present study. The Tiger corpus (Brants et al., 2004)² contains about 50k sentences with about 888k tokens from 2,263 German news articles, but the annotation of certain syntactic phenomena deviates significantly from those in the TüBa-D/Z corpus (Dipper and Kübler, 2017). Most notably, the Tiger treebank includes discontinuous annotations. Therefore, all sentences must be linearized first³ before chunks of the six different types can be extracted from the constituency trees similar to the procedure described by Kübler et al. (2010).

Besides accounting for possible influences of the annotation scheme on the extracted chunks, including the Tiger treebank offers another advantage: While annotated historical data sets rarely exist for syntactic annotation tasks, there are two

¹Release 11.0, chunked version, <http://www.sfs.uni-tuebingen.de/ascl/ressourcen/corpora/tueba-dz.html>

²Version 2.2, TIGER-XML format, <https://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger>

³As only the lowest phrasal projections are used to derive chunks from the tree, the broader structure of the tree is irrelevant for the task at hand. Therefore, discontinuous nodes are simply duplicated and re-inserted at the correct position inside the tree according to the linear order of terminal nodes in the sentence.

Corpus	#Docs	#Sents	#Toks	#Chunks
<i>Training</i>				
TüBa-D/Z	3,075	83,225	1,564,840	593,735
Tiger	1,863	39,976	726,811	255,077
Mercurius	2	6,709	150,354	53,831
ReF.UP	26	16,761	415,934	163,438
<i>Development</i>				
TüBa-D/Z	377	10,702	196,308	74,780
Tiger	200	4,567	81,593	28,615
Mercurius	2	820	18,287	6,570
ReF.UP	26	2,112	53,836	21,245
<i>Test</i>				
TüBa-D/Z	364	10,491	196,636	74,982
Tiger	200	4,445	78,018	27,253
Modern	78	547	7,605	2,829
Mercurius	2	818	18,740	6,691
ReF.UP	26	2,173	54,005	21,120
HIPKON	53	342	4,210	1,529
DTA	29	606	18,515	6,651

Table 1: Overview of the data sets. The number of chunks refers to the six chunk types evaluated in this study. Only sentences containing at least one chunk of the given types are included.

treebanks for historical German, which are annotated according to the Tiger scheme and thus, fortunately, can also be used for chunk extraction. The Mercurius corpus (Demske, 2005)⁴ contains semi-automatic annotations of approximately 8k sentences with 187k tokens from newspaper text from the 16th and 17th centuries. The second treebank, ReF.UP, is a subcorpus of the Reference Corpus of Early New High German (Wegera et al., 2021)⁵ and includes annotations of 26 documents with 21k sentences and 500k tokens from different language areas from the 14th to 17th century. Like with the Tiger corpus, the constituency trees from both historical treebanks must be linearized before chunks can be extracted from them. In total, the two corpora contain about 67k chunks and over 205k chunks of the six relevant types, respectively. While the Tiger corpus is already provided with a training, development, and test section, the other three corpora were split into a training (80%), development (10%), and test set (10%) for this study. Also, the historical POS tagsets in the Mercurius and ReF.UP treebanks were mapped to the German standard tagset STTS (Schiller et al., 1999).

Compared to previous studies on historical data, the two modern and historical treebanks form a solid basis for training and evaluating automatic

⁴Mercurius Baumbank (version 1.1), <https://doi.org/10.34644/laudatio-dev-VyQiCnMB7CArCQ9cJF30>

⁵<https://www.linguistics.rub.de/ref>

chunking methods on historical German. However, Osborne (2002) notes that distributional differences between training and test data can be even more problematic for chunking performance than noise in the data itself. Therefore, three additional data sets from a previous study (Ortmann, 2020),⁶ which are unrelated to the training data, are used for evaluation. The first data set is a collection of about 550 sentences with 7.6k tokens from five modern registers with a varying degree of formality: Wikipedia articles, fiction texts, Christian sermons, TED talk subtitles, and movie subtitles. In total, the modern data set contains about 2.8k chunks of the six types and is used to test the applicability of annotation methods to non-newspaper registers.

The two other data sets comprise historical data from two different corpora. The HIPKON corpus (Coniglio et al., 2014) contains 342 manually annotated sentences from 53 sermons from the 12th to the 18th century. Originally, the corpus only includes a partial annotation of chunks, which was completed for the present study. Also, the mapping of the historical POS tags to STTS tags from Ortmann (2020) was used. The second historical data set consists of 600 sentences with 18.5k tokens from 29 texts from the German Text Archive DTA (BBAW, 2021). The texts were published in a variety of genres⁷ from the 16th to the 20th century and were manually enriched with chunks for this study, using the corrected POS tags and sentence boundaries from Ortmann (2020). Table 1 gives an overview of the data sets. The annotated data sets and additional resources can be found in this paper’s repository.⁸

Table 2 shows the distribution of the six chunk types in the test data. As could be expected, noun chunks (NC) are the most frequent chunk type, followed by prepositional chunks (PC) and adverb chunks (ADVC). Stranded chunks make up about 1% of the chunks in all data sets, except for the TüBa-D/Z data with 0.6% and the modern non-standard data with only 0.4% stranded chunks. While stranded noun chunks (sNC) are more frequent in the modern data, the opposite can be observed for most of the historical data sets where

⁶<https://github.com/rubcompling/latech2020>

⁷The DTA subset contains five newspaper texts and three texts each from the genres: funeral sermon, language science, medicine, gardening, theology, chemistry, law, and prose.

⁸<https://github.com/rubcompling/nodalida2021>

Corpus	NC	PC	AC	ADVC	sNC	sPC
TüBa-D/Z	54.2	24.6	5.9	14.8	0.4	0.2
Tiger	55.2	30.7	4.6	8.5	0.6	0.4
Modern	60.3	21.2	5.5	12.5	0.3	0.1
Mercurius	51.5	29.5	4.4	13.5	0.4	0.7
ReF.UP	57.7	20.6	5.9	15.1	0.2	0.5
HIPKON	56.4	25.1	2.4	15.3	0.1	0.9
DTA	56.4	24.4	5.2	12.8	0.6	0.6

Table 2: Distribution of chunk types in the test data reported as percentage of the total number of chunks per data set.

stranded prepositional chunks (sPC), as in example (6) from the Mercurius corpus, are more common.

- (6) [sPC von] [NC der Frantzosen] [PC Vorhaben]
of the French’s plan
‘of the plan of the French’

5 Methods

As detailed in Section 3, various methods have been applied to the automatic recognition of chunks in modern text. In the present study, two different approaches are tested: an unlexicalized regular expression-based chunker, which serves as a baseline, and a neural state-of-the-art sequence labeling tool.

The regular expression-based approach is comparable to the finite-state chunkers mentioned in Section 3. For this study, a simple RegExp chunker as implemented in the NLTK⁹ is used, which successively applies a set of manually created context-sensitive regular expressions to an input POS sequence to identify non-recursive, non-overlapping chunks of the six different types.

The neural sequence labeling tool NCRF++ (Yang and Zhang, 2018)¹⁰ achieves state-of-the-art results for several tasks, including chunking. On the English CoNLL-2000 data, the best model reaches an F_1 -score of 95% (Yang et al., 2018). The toolkit consists of a three-layer architecture with a character sequence layer, a word sequence layer, and a CRF-based inference layer. While the RegExp chunker relies on expert knowledge in the form of manually compiled rules, NCRF++ must be trained on annotated data to perform the task. For this study, the tool is trained on the two different modern treebanks: model News1 is trained

⁹<http://www.nltk.org/api/nltk.chunk.html>

¹⁰<https://github.com/jiesutd/NCRFpp>

on the TüBa-D/Z training set, and model News2 on the Tiger training set. Also, the two historical treebanks are used to train a joined model Hist, which might be more suitable for the analysis of historical data and its peculiarities. Finally, since the historical data sets are smaller than the modern training sets, a model News2+Hist is trained on a combination of the modern and historical treebanks that follow the same annotation scheme. During training, the tool is provided with the corresponding development data and each of the models is trained with and without POS tags as an additional feature. Since current contextual word representations seem to be aware of shallow syntactic structures (Swayamdipta et al., 2019), each model is also trained with GloVe embeddings pre-trained on German Wikipedia.¹¹ To ensure comparability, all models are trained with the same default settings.¹² While the News2 and Hist training sets only contain annotations of the six chunk types considered in this study, the News1 model is trained on all chunk types included in the TüBa-D/Z corpus, although only the six types described in Section 2 are evaluated here. For each token, both selected methods, i.e. the RegExp chunker and the NCRF++ toolkit, output the single most likely chunk label encoded as a BIO tag.

6 Evaluation and Results

To assess the performance of the automatic methods introduced in the previous section, their output is compared to the gold standard annotation. As already mentioned, every token is annotated with a BIO tag, i.e. either B-XC (beginning of chunk), I-XC (inside chunk), or O (outside chunk). However, the number of tokens inside and outside of chunks provides little information about the quality of the automatic chunk annotation. Instead, it is of interest whether the boundaries of chunks align between automatic and gold annotation. Therefore, the evaluation is carried out chunk-wise instead of token-wise and each chunk in the gold

¹¹GloVe embeddings trained on German Wikipedia and provided by deepset, <https://deepset.ai/german-word-embeddings>

¹²The experiments of Yang et al. (2018) suggest that the default combination of character CNN, word LSTM, and a CRF-based inference layer gives the best result for the chunking task with good model stability for random seeds (mean F_1 : 94.86 ± 0.14). However, the present study is only a first investigation of chunking historical German and further experiments should be conducted to test for model stability and to explore fine-tuning of parameters for optimal results.

Model	Words	POS	GloVe	TüBa-D/Z	Tiger	Modern	Mercurius	ReF.UP	HIPKON	DTA
RegExp	-	+	-	85.46	86.75	90.35	85.70	86.83	91.76	88.20
News1	+	-	-	93.46	87.80	89.63	72.52	49.77	47.69	72.07
	+	-	+	94.30	88.16	90.12	73.48	51.94	48.43	71.50
	+	+	-	97.07	90.33	92.91	90.34	91.01	93.71	90.11
	+	+	+	97.17	90.89	93.68	90.37	90.66	92.92	90.15
News2	+	-	-	85.02	91.41	86.67	71.15	49.09	43.25	67.75
	+	-	+	86.19	92.76	87.77	72.05	50.01	46.90	69.59
	+	+	-	90.96	94.70	94.04	88.58	89.84	94.20	88.76
	+	+	+	91.22	95.44	93.97	88.55	88.77	92.50	88.35
Hist	+	-	-	n.a.	n.a.	n.a.	11.68	16.10	12.81	13.86
	+	-	+	n.a.	n.a.	n.a.	85.53	81.28	69.41	73.61
	+	+	-	n.a.	n.a.	n.a.	92.37	93.48	93.29	89.89
	+	+	+	n.a.	n.a.	n.a.	92.80	93.64	93.85	90.37
News2 +Hist	+	-	-	n.a.	n.a.	n.a.	82.56	79.42	60.47	73.24
	+	-	+	n.a.	n.a.	n.a.	83.40	79.02	65.05	74.77
	+	+	-	n.a.	n.a.	n.a.	91.94	93.03	94.49	90.15
	+	+	+	n.a.	n.a.	n.a.	92.19	93.41	93.99	90.29

Table 3: Overall F_1 -scores for the RegExp chunker and all NCRF++ models for the seven corpora. Models trained on historical data are only applied to historical corpora. All numbers are given in percent and the best result for each corpus is highlighted in bold.

standard is compared to the system output and vice versa concerning chunk type and chunk boundaries. Only sentences for which the gold standard contains at least one of the six relevant chunk types are considered. Chunks with identical labels and boundaries are counted as true positives, whereas chunks only existing in the gold standard are considered false negatives, and chunks only present in the system output count as false positives.

In addition to these common categories, there can be additional types of errors, though, which are not captured by the three categories and usually are penalized as multiple errors in a single unit. For example, a system could identify a chunk spanning the correct token sequence but label it as a different chunk type, e.g. *ADVC* instead of *AC*, which would count as a false positive *ADVC* and a false negative *AC*. Also, a system can get the boundaries of a chunk wrong, e.g. miss the first word of an *ADVC*, which would correspond to a false positive and a false negative *ADVC*. And finally, the system can make both errors at once, for example by missing the initial preposition and classifying a *PC* as *NC*, resulting in a false positive *NC* and a false negative *PC*. To account for these types of errors, in the following, seven different categories are distinguished during evaluation: true positives (TP), false positives (FP), labeling errors (LE), boundary errors (BE), labeling-boundary errors (LBE), and false negatives (FN).¹³

¹³The idea for this distinction between error types stems from a blog post by Chris Manning about

Because labeling and boundary errors mean that the system recognized some chunk, although not entirely correctly, and not that it missed a chunk, LE, BE, and LBE errors are counted as false positives for the calculation of precision and recall while preventing multiple penalties for a single unit. As the evaluation is carried out chunk-wise, sensible true negatives cannot be determined and are not evaluated here. Table 3 gives an overview of the results for the different annotation methods and models.

The evaluation shows that the RegExp parser, which operates on POS tags only, reaches F_1 -scores between 85% and 92% for all data sets, setting a high baseline for the task. The best results are achieved for the modern non-newspaper data and the HIPKON corpus. The NCRF++ models outperform this baseline by several percentage points on each data set, achieving F_1 -scores between 90% and 97%. The recall lies between

a similar problem with named entity evaluation (<https://nlpers.blogspot.com/2006/08/doing-named-entity-recognition-dont.html>). The problem with double penalties when using F-scores has also been recognized in the literature. For example, in the context of word tokenization, Shao et al. (2017) show that precision favors under-splitting systems, suggesting that recall, i.e. the proportion of correctly segmented units, gives a more realistic impression of system performance and should be used as the only evaluation metric. However, for tasks that require segmentation and labeling such as chunking or NER, almost correct chunks/entities may still provide useful information for certain purposes. Thus, the more fine-grained distinction of errors and adjusted calculation of precision and recall seem appropriate for a thorough evaluation of these annotations.

97% and 99% for the best models on all data sets and is always higher than the precision with 84% to 95%. As already observed in other studies (van den Bosch and Buchholz, 2002), models that include POS as additional features generally perform better than models purely based on characters and word forms. Also, adding pre-trained word embeddings improves the results in almost all cases, especially for models without POS tags.

The modern newspaper data is analyzed with the highest F_1 -scores of 97% and 95% respectively. Unsurprisingly, models trained on the training section of the same corpus perform better on the test data than models trained on another data set. This may be a result of distributional differences between data sets (Osborne, 2002) but could, in part, also be due to differences between the constituency trees from which the chunks were extracted.

The results for the modern non-newspaper data are slightly lower than for the news corpora with a maximum F_1 -score of 94%. Interestingly, the overall F_1 -scores are higher for the more informal registers than for the formal ones. Probably, informal sentences are generally easier to chunk because they contain more simple (noun) chunks and less pre-nominal modification.

While models purely based on words still perform well on the modern data, POS tags prove to be especially relevant for the historical data. Even the `Hist` model must be complemented with (modern) pre-trained word embeddings for acceptable performance on the historical corpora, possibly reflecting problems with the non-standardized spelling in historical German. For the `Mercurius` and `ReF.UP` corpora, the `Hist` model with POS and word embeddings achieves the best results with F_1 -scores of about 93%, followed by the `News2+Hist` model. For the `HIPKON` corpus, the `News2+Hist` model with POS reaches the highest F_1 -score of 94.5%, closely followed by the `News2` model. The `DTA` data is analyzed with the highest F_1 -score of 90.4% by the `Hist` model with POS and word embeddings, followed by the `News2+Hist` and the `News1` models with F_1 -scores of about 90% as well.

These results are in line with the observations of Ortmann (2020) that models trained on modern news data can successfully be transferred to historical German with overall F_1 -scores $>90\%$ when POS tags are used as input. However, the

Corpus	NC	PC	AC	ADVC	sNC	sPC
TüBa-D/Z	95.6	97.9	86.8	97.0	77.2	70.0
Tiger	94.4	95.0	85.2	84.7	84.7	68.4
Modern	93.3	91.3	85.4	83.7	80.0	0.0
Mercurius	90.6	90.8	84.3	86.0	0.0	36.7
ReF.UP	92.9	92.3	81.1	85.1	5.6	40.3
HIPKON	94.1	90.4	87.0	87.4	0.0	26.7
DTA	87.5	90.0	80.4	81.8	10.3	16.7

Table 4: Overall F_1 -scores per chunk type (in percent) for the best performing model on each data set.

evaluation also shows that historical training data further improves the automatic annotation of historical language.¹⁴

In Table 4, the results per chunk type are displayed for the best performing model on each data set. Here, no distinction is made between true positives, labeling, and boundary errors, i.e. one unit can correspond to multiple errors in one or two of the categories as exemplified above. For all data sets, the best results are observed for noun and prepositional chunks with F_1 -scores mostly above 90%, while the results for adjective and adverb chunks range mostly between 80% and 87%. The stranded chunk types are recognized much less reliably, especially in the historical data where the majority of errors in these categories result from structures with a pre-nominal modifying noun chunk `NC` inside a prepositional chunk `PC` like in example (6) above. These structures are more frequent in historical German, causing the higher proportion of stranded prepositional chunks compared to modern data. When confronted with a structure like this, in most cases, instead of annotating a stranded preposition `sPC` preceding a pre-nominal noun chunk `NC`, the models identify a joined `PC`, followed by an `NC` as in example (7).

¹⁴It is important to note that the experiments in this paper were conducted with gold standard POS tags and using automatically assigned POS can be expected to negatively influence the results. For example, Müller (2005) reports a chunking F_1 -score of only 90% instead of 96% when using automatic POS. Applying the Stanza tagger (Qi et al., 2020, German `hdt` model) to the modern data sets in this study results in POS error rates of 4% (TüBa-D/Z) to 6% (Modern) and reduces the F_1 -scores of the `RegExp` chunker by 1 (TüBa-D/Z) to 4 (Modern) percentage points. The F_1 -scores of the best `NCRF++` models with POS as feature decrease by 3 (TüBa-D/Z) to 3.7 (Tiger, Modern) percentage points. It can be assumed that similar reductions would be observed for historical data if a comparable tagger model for the relevant language stages was available and used to tag the data automatically.

Corpus	FP	LE	BE	LBE	FN
TüBa-D/Z	10.9	4.4	60.1	4.8	19.8
Tiger	17.7	6.0	59.8	4.2	12.3
Modern	11.3	5.5	63.6	2.4	17.1
Mercurius	22.6	10.3	53.2	7.1	6.7
ReF.UP	17.5	8.1	56.0	7.1	11.3
HIPKON	11.7	10.4	55.8	12.3	9.8
DTA	13.9	6.5	58.9	6.7	14.0

Table 5: Proportion of the five different error types: false positives (FP), labeling errors (LE), boundary errors (BE), labeling-boundary errors (LBE), and false negatives (FN). Numbers are given in percent for the best performing model on each data set.

- (7) **Gold:** [sPC von] [NC der Frantzosen] [PC Vorhaben]
NCRF++: [PC von der Frantzosen] [NC Vorhaben]

Since, in these cases, the embedded noun chunk cannot be recognized based on STTS POS tags, a morphological analysis is necessary to distinguish structures with a pre-nominal genitive from prepositional chunks with a post-modifying noun chunk. When the genitive form is not syncretized, i.e. the word form differs from the morphological realization in other cases like nominative or dative, lexicalized models could, in theory, identify the correct structure. But as stranded chunks constitute only about one percent of all chunks in the data sets, there is not enough training data to recognize them reliably.

Finally, Table 5 shows the distribution of error types in the data sets, including the more fine-grained distinction of labeling and boundary errors. Interestingly, for all corpora, boundary errors constitute more than half of the errors, i.e. the models identified the chunks but did not achieve an exact match of the boundaries. One could argue that this type of error is less severe than completely missing (FN) or made-up chunks (FP), which are the second and third most frequent error types for most data sets. The evaluation approach in this study, which does not multiply penalize a model for boundary errors, thus seems appropriate to get a more realistic impression of model performance.

7 Conclusion

The present study has investigated the automatic recognition of chunks in historical German. To address the main problem of analyzing historical language, namely a lack of manually annotated data

for training and evaluation, chunks of six different types were derived from modern and historical constituency treebanks. Using the extracted chunks, the state-of-the-art neural sequence labeling tool NCRF++ was trained on modern news articles, Early New High German corpora, as well as a combination of modern and historical data.

The evaluation has shown that models that include POS tags as features can be transferred successfully from modern to historical language, with F_1 -scores $>90\%$, thereby outperforming a regular expression-based baseline. By adding historical training data, the results can be improved further, yielding F_1 -scores between 90.4% and 94.5% for the different historical corpora.

Regarding the evaluation of chunks, the present study has argued for a distinction between different types of errors that are commonly penalized as multiple errors in a single unit. An analysis of the occurring error types showed that the majority of errors are boundary errors, meaning that the system identified the chunks, but the boundaries do not exactly match those in the gold standard. Since this type of error can be considered less severe than pure false positives or negatives, the presented results give a more realistic impression of the actual system performance.

Future studies should focus primarily on a reduction of incorrect chunk boundaries to increase the annotation precision, as well as further investigate and improve the analysis of stranded chunks and complex pre-nominal modification in (historical) German.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102 (Project C6). I am grateful to the student annotators Anna Maria Schroeter and Larissa Weber for the annotations and Jennifer Wodrich for help with the various data sets. Also, I would like to thank the anonymous reviewers for their helpful comments.

References

- Steven P. Abney. 1991. Parsing by chunks. In Robert C. Berwick, Steven P. Abney, and Carol Tenny, editors, *Principle-based parsing*, volume 44 of *Studies in Linguistics and Philosophy*, pages 257–278. Springer.

- Adnan Akhundov, Dietrich Trautmann, and Georg Groh. 2018. Sequence labeling: A practical approach. *arXiv preprint arXiv:1808.03926*.
- BBAW. 2021. Deutsches Textarchiv. Grundlage für ein Referenzkorpus der neuhochdeutschen Sprache. Berlin-Brandenburgische Akademie der Wissenschaften.
- Antal van den Bosch and Sabine Buchholz. 2002. Shallow parsing on the basis of words only: a case study. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 433–440.
- Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkor-eit. 2004. TIGER: Linguistic interpretation of a German corpus. *Research on language and computation*, 2(4):597–620.
- Christian Chiacos, Benjamin Kosmehl, Christian Fäth, and Maria Sukhareva. 2018. Analyzing Middle High German syntax with RDF and SPARQL. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Marco Coniglio, Karin Donhauser, and Eva Schlachter. 2014. HIPKON: Historisches Predigtenkorpus zum Nachfeld (Version 1.0). Humboldt-Universität zu Berlin. SFB 632 Teilprojekt B4.
- Michael Daum, Kilian A. Foth, and Wolfgang Menzel. 2003. Constraint based integration of deep and shallow parsing techniques. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary.
- Ulrike Demske. 2005. Mercurius-Baumbank (Version 1.1). Universität Potsdam.
- Stefanie Dipper, Hannah Kermes, Dr. Esther König-Baumer, Wolfgang Lezius, Frank H. Müller, and Tylman Ule. 2002. DEREKO (DEutsches REferenzKOrpus) German Reference Corpus Final Report (Part I).
- Stefanie Dipper and Sandra Kübler. 2017. German treebanks: TIGER and TüBa-D/Z. In Nancy Ide and James Pustejovsky, editors, *Handbook of linguistic annotation*, pages 595–639. Springer.
- Gerhard Fliedner. 2002. A system for checking NP agreement in German texts. In *Proceedings of the ACL Student Research Workshop*, pages 12–17.
- Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.
- Balázs Indig. 2017. Less is more, more or less... Finding the optimal threshold for lexicalization in chunking. *Computación y Sistemas*, 21(4):637–646.
- Sandra Kübler, Kathrin Beck, Erhard Hinrichs, and Heike Telljohann. 2010. Chunking German: an unsolved problem. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 147–151, Uppsala, Sweden. Association for Computational Linguistics.
- Cerstin Mahlow and Michael Piotrowski. 2010. Noun phrase chunking and categorization for authoring aids. In *10. Konferenz zur Verarbeitung Natürlicher Sprache (KONVENS 2010)*. University of Zurich.
- Antonio Molina and Ferran Pla. 2002. Shallow parsing using specialized HMMs. *The Journal of Machine Learning Research*, 2:595–613.
- Frank Henrik Müller. 2005. *A finite-state approach to shallow parsing and grammatical functions annotation of German*. Ph.D. thesis, Seminar für Sprachwissenschaft, Universität Tübingen.
- Katrin Ortmann. 2020. Automatic Topological Field Identification in (Historical) German Texts. In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 10–18.
- Miles Osborne. 2000. Shallow parsing as part-of-speech tagging. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*, pages 145–147.
- Miles Osborne. 2002. Shallow parsing using noisy and non-stationary training material. *The Journal of Machine Learning Research*, 2(Mar):695–719.
- Petya Osenova and Kiril Simov. 2003. Between chunk ideology and full parsing needs. In *Proceedings of the Shallow Processing of Large Corpora (SProLaC 2003) Workshop*, pages 78–87.
- Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*.
- Florian Petran. 2012. Studies for segmentation of historical texts: Sentences or chunks? In *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2)*, pages 75–86.
- Alexandre Pinto, Hugo Gonçalo Oliveira, and Ana Oliveira Alves. 2016. Comparing the performance of different NLP toolkits in formal and social media text. In *5th Symposium on Languages, Applications and Technologies (SLATE'16)*, pages 3:1–3:16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

- Luzia Roth and Simon Clematide. 2014. Tagging complex non-verbal German chunks with Conditional Random Fields. In *Proceedings of the 12th Edition of the KONVENS Conference*, pages 48–57, Hildesheim, Germany. University of Zurich.
- Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*, pages 127–132.
- Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset).
- Yan Shao, Christian Hardmeier, and Joakim Nivre. 2017. Recall is the proper evaluation metric for word segmentation. In *Proceedings of the The 8th International Joint Conference on Natural Language Processing*, pages 86–90, Taipei, Taiwan.
- Hong Shen and Anoop Sarkar. 2005. Voting between multiple data representations for text chunking. In Balázs Kégl and Guy Lapalme, editors, *Advances in Artificial Intelligence. Canadian AI 2005.*, pages 389–400. Springer.
- Xu Sun, Louis-Philippe Morency, Daisuke Okanohara, Yoshimasa Tsuruoka, and Jun’ichi Tsujii. 2008. Modeling latent-dynamic in shallow parsing: a latent conditional model with improved inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 841–848, Manchester, UK.
- Swabha Swayamdipta, Matthew Peters, Brendan Roof, Chris Dyer, and Noah A. Smith. 2019. Shallow syntax in deep water. *arXiv preprint arXiv:1908.11047*.
- Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. 2017. *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Seminar für Sprachwissenschaft, Universität Tübingen, Germany.
- Vincent Van Asch and Walter Daelemans. 2009. Prepositional phrase attachment in shallow parsing. In *Proceedings of the International Conference RANLP-2009*, pages 12–17. Association for Computational Linguistics.
- Klaus-Peter Wegera, Hans-Joachim Solms, Ulrike Demske, and Stefanie Dipper. 2021. Referenzkorpus Frühneuhochdeutsch (Version 1.0).
- Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 3879–3889, Santa Fe, New Mexico, USA.
- Jie Yang and Yue Zhang. 2018. NCRF++: An open-source neural sequence labeling toolkit. In *Proceedings of ACL 2018, System Demonstrations*, pages 74–79, Melbourne, Australia.
- Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. 2017. Neural models for sequence chunking. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3365–3371.

Part-of-speech tagging of Swedish texts in the neural era

Yvonne Adesam and Aleksandrs Berdicevskis

Språkbanken Text

Department of Swedish

University of Gothenburg

yvonne.adesam@gu.se, aleksandrs.berdicevskis@gu.se

Abstract

We train and test five open-source taggers, which use different methods, on three Swedish corpora, which are of comparable size but use different tagsets. The KB-Bert tagger achieves the highest accuracy for part-of-speech and morphological tagging, while being fast enough for practical use. We also compare the performance across tagsets and across different genres. We perform manual error analysis and perform a statistical analysis of factors which affect how difficult specific tags are. Finally, we test ensemble methods, showing that a small (but not significant) improvement over the best-performing tagger can be achieved.

1 Introduction

The standard approach to automatic part-of-speech tagging for Swedish has been using the Hunpos tagger (Halácsy et al., 2007), trained by Megyesi (2009) on the Stockholm-Umeå corpus (Ejerhed et al., 1992). Just over a decade later neural methods have reshaped the NLP landscape, and it is time to re-evaluate which taggers are most accurate and effective for Swedish text.

In this paper we explore part-of-speech and morphological tagging for Swedish text. The primary purpose is to see which tagger or taggers to include in the open annotation pipeline Sparv¹ (Borin et al., 2016) for tagging the multi-billion token corpora of Språkbanken Text, available through Korp² (Borin et al., 2012). We therefore train and test a set of part-of-speech taggers, which rely on different methods, on a set of corpora of comparable size, with different part-of-

speech annotation models. We apply a 5-fold training and evaluation regime.

In Section 2 we describe the corpora, and in Section 3 the taggers and models. We evaluate the taggers along a number of dimensions in Section 4, including the potential for using ensemble methods, and discuss the results in Section 5.

2 Data

2.1 Corpora and tagsets

Corpora and treebanks have a long history in Sweden; the first large annotated treebank, Talbanken, was compiled in the mid 1970s (Teleman, 1974). For several decades, the Stockholm-Umeå corpus (SUC, Ejerhed et al., 1992) has been the main resource for training part-of-speech taggers.

In this paper, however, we use three other corpora: Talbanken-SBX, Talbanken-UD, and Eukalyptus. The primary reason for using these three resources is that they are annotated with different tagsets, which allows us to compare results between tagsets. Talbanken-SBX follows the same annotation model as SUC. Talbanken-UD follows the Swedish version of the Universal Dependencies (UD) framework (Nivre et al., 2016; Nivre, 2014). The UD project develops a cross-linguistic annotation framework and resources annotated with it for a large number of languages. In contrast, the Eukalyptus treebank (Adesam et al., 2015) was developed specifically for Swedish to be “in line with the currently standard view on Swedish grammar” (Adesam and Bouma, 2019, p. 7). We also exclude SUC because these three resources are of comparable size – close to 100,000 tokens and with a type-token ratio of around 0.17. SUC is much larger, and would have to be scaled down to be comparable.

We briefly describe the corpora below. For consistency, we use the same terms to describe the annotation in the corpora: POS for coarse-

¹<https://spraakbanken.gu.se/sparv>

²<https://spraakbanken.gu.se/korp>

	TB-SBX	TB-UD	Euk
Tokens	96,346	96,858	99,909
Types	16,242	16,305	17,237
POS-tags	25	16	13
MSD-tags	130	213	117

Table 1: Statistics for the corpora used in the tagging experiments; Talbanken-SBX, Talbanken-UD, and Eukalyptus. Tag counts are used tags, not potential tags.

grained part-of-speech level tags and MSD for finer-grained morphosyntactic descriptions (*features* in the UD parlance).

The two Talbanken corpora originate from a subset (the professional prose section) (Nivre et al., 2006) of the original Talbanken (Teleman, 1974), which was converted to the SUC tagset (Ejerhed et al., 1992) for the Swedish Treebank (Nivre and Megyesi, 2007)³. The morphological annotation was manually checked and revised. Both Talbanken-SBX and Talbanken-UD are based on the output of this conversion.

*Talbanken-SBX*⁴ has the converted SUC tags, and is the result of some minor corrections made later at Språkbanken Text. Among our three corpora, the SUC tagset is the largest set at the POS-level (see Table 1). It has a very fine-grained set of tags for determiners, pronouns, adverbs, and punctuation symbols. There are also separate tags for infinitival markers, participles, verb particles, and ordinals.

*Talbanken-UD*⁵ is the result of an independent conversion of the same corpus to UD. The texts themselves were cleaned during this conversion, some sentences that had been lost during the initial conversion were recovered, and sentence segmentation and the order of texts was changed. Thus, Talbanken-UD and Talbanken-SBX are not strictly parallel. The conversion to UD has partly been manually checked and revised. We use version 2.7.

The number of POS-tags in the UD tagset is quite small, but together with MSD-tags the tagset

³https://cl.lingfil.uu.se/~nivre/swedish_treebank/

⁴<https://spraakbanken.gu.se/en/resources/talbanken>

⁵https://universaldependencies.org/treebanks/sv_talbanken/index.html

is the largest among our corpora (Table 1). The tagset does not have separate categories for the infinitival marker, ordinals, or participles. It also does not mark foreign words as a category, but instead treats this as a feature in the morphological description. In contrast to the other tagsets, it does, however, mark auxiliaries separately.

*Eukalyptus*⁶ contains texts of five different types, including Wikipedia and blog texts, which makes this data the most recent and allows us to compare different genres. The tagset loosely builds upon the SUC tagset. The treebank is currently in an early version, and although tagging has been checked, there are still some known errors, such as inconsistencies in noun gender. This tagset is the smallest one, both at POS- and MSD-levels (Table 1). The tagset does not, for example, distinguish determiners, infinitival markers, participles, particles, or ordinals as separate categories.

2.2 Preprocessing and data splits

We pre-processed all corpora in a similar manner. For all corpora, spaces within tokens, if present, were replaced with underscore, since some taggers do not allow spaces in the input. We divided all three datasets into five folds for cross-validation. In the case of Eukalyptus, the treebank is shipped in five different files, one for each text type, which were used as is. In the case of Talbanken, we split the data into five consecutive splits, i.e. putting the first fifth of the data into the first split, the second fifth in the second, etc. We would have preferred to divide the data according to text types or documents, but this is not easily retrievable for all the data. Using consecutive splits rather than random splits or splits where the first sentence is put in the first split, the second sentence in the second split, etc, means that the data splits are more distinct than with random splits (see the discussion in e.g. Gorman and Bedrick, 2019; Søggaard et al., 2020). This means that the same text is not divided over all splits, although possibly into two splits.

One of the five folds (20%) is always used a test set. Some of the taggers we investigated do use a separate validation (dev) set, some do not (see Table 2). For the latter ones, we merge all four remaining folds into a training set (80%). For the former ones, we first merge the four folds and then

⁶<https://spraakbanken.gu.se/en/resources/eukalyptus>

randomly (not consecutively) split them into train and dev in the proportion 3:1 (60% of the total data for training and 20% for validation). We consider this solution to be more fair to the “dev-less” taggers than using the same training sets throughout and then adding dev for some taggers, but not for others.

3 Taggers

We have selected five open-source taggers. Our goal was to sample taggers that use different methods, are (or were at some point) known to have high performance and either can be easily incorporated into our annotation pipeline Sparv or already are (as Hunpos and Stanza). This last consideration steers the selection to a large extent (Stanza, for instance, has an important advantage of being a convenient pipeline that achieves high performance on other tasks, such as dependency parsing).

We also wanted to compare taggers that were state-of-the-art in the “pre-neural” era⁷ with the current ones. The key properties of the taggers are summarized in Table 2. Note that the classification in the “Key method” column is of course very crude (Flair, for instance, can be labelled as both neural and CRF).

As can be seen from the table, different taggers use different kinds of additional information. Hunpos does not take any further input. For Marmot, we plug in Saldo-Morphology (Borin et al., 2013), a morphological dictionary of 1M words with a tagset that is similar (but not equivalent) to the SUC tagset. From previous experiments we know that using Saldo gives Marmot a boost when it is applied to texts tagged with the SUC tagset (i.e. TalbankenSBX in our case). We assume it can also boost performance on Eukalyptus, since the tagsets are similar, but we do not expect a boost for UD. For Stanza, we use word2vec embeddings⁸ trained on the CONLL17 corpus (Zeman et al., 2017), which was built using the CommonCrawl data and contains approximately 3 billion words for Swedish. One of the main ideas of Flair is to combine various types of embeddings; the best combination we were able to find was that of the CONLL17 word2vec and Flair’s

⁷An anonymous reviewer notes that the best label for the current era is not “neural”, but “post-neural” or “language-model” era.

⁸<http://vectors.nlpl.eu/repository>

own embeddings (trained on Wikipedia/OPUS⁹, size is not reported). For KB-Bert¹⁰, we used the `bert-base-swedish-cased` model, trained by the Datalab of the National Library of Sweden (KB) on 3.5 billion words from the library collection. The collection contains predominantly (85%) newspaper texts, but also official reports from authorities, books, magazines, social media and Wikipedia. The training and tagging itself was done as in (Malmsten et al., 2020), using the `runner.py` script from the Huggingface framework¹¹. For Stanza and Flair, we experimented with using different classic and contextualized embeddings, for instance, word2vec trained on a press corpus (Fallgren et al., 2016) or Bert instead of Flair’s own embeddings, but the results were always slightly worse than those we report.

4 Evaluation

We evaluate the taggers on the treebanks along several dimensions. In the following we report tagger speed and accuracy. We also explore unseen words, specific tags that seem more difficult to get right, as well as an ensemble approach.

4.1 Speed

We trained the neural taggers on GPU (on CPU the training time is prohibitively long) and the non-neural ones on CPU. This means the time measurements are not directly comparable, and we thus do not report detailed quantitative results, but the qualitative picture is very clear. For Hunpos, the training on one fold takes about a second, so does tagging. For Marmot, training takes about 1.5 minutes, tagging about 10 seconds. For Stanza, training takes about 2 hours, tagging about 8 seconds. For Flair, training takes about 6 hours, tagging about 5 seconds. KB-Bert, however, breaks the pattern “the better the slower”: training takes about 3 minutes, tagging takes about 5 seconds. Note that for the neural taggers the tagging time

⁹https://github.com/flairNLP/flair/blob/master/resources/docs/embeddings/FLAIR_EMBEDDINGS.md

¹⁰The script crashes if the dev set contains previously unseen tags. To solve this, we replace all such tags with the tag for adverb (AB for SBX and Eukalyptus, ADV for UD) when training Bert. This can potentially affect the results, but the number of such tags is always small (varying from 0 to 10 across various folds), which should only give a negligible bias against KB-Bert.

¹¹<https://github.com/huggingface/transformers/blob/master/examples/token-classification>

Name	Key method	Embeddings		Dictionary	Dev	References
		Token	Type			
KB-Bert	Neural	KB-BERT	-	-	Yes	Malmsten et al. (2020); Wolf et al. (2020)
Flair	Neural	Flair	Word2vec	-	Yes	Akbik et al. (2019)
Stanza	Neural	-	Word2vec	-	Yes	Qi et al. (2020)
Marmot	CRF	-	-	SALDO	No	Mueller et al. (2013)
Hunpos	HMM	-	-	-	No	Halácsy et al. (2007)

Table 2: Basic info about the taggers. HMM = hidden Markov models, CRF = conditional random fields, Dev = whether the tagger uses a development set. Type embeddings = “classic” (“static”) embeddings, token = “contextualized” (“dynamic”).

excludes the time necessary to load models, embeddings and all necessary modules. If this is taken into account, the tagging time becomes considerably longer (for KB-Bert, for instance, about 30 seconds).

4.2 Overall tagging quality

Table 3 shows the accuracy (macroaverage over 5 folds) for the full POS+MSD label. It shows that KB-Bert achieves the best results, and that the Talbanken-SBX corpus is easiest to tag, while Eukalyptus has lower results. It is not surprising that the newer neural models perform the best, while the older models achieve lower scores. To test whether differences between the taggers are significant, we rank them by performance and then do pairwise comparisons of adjacent taggers (KB-Bert and Flair, Flair and Stanza etc.) by running paired two-tailed *t*-tests on 15 (3x5) datapoints. We apply the same procedure to the sentence-level accuracy (Table 5) and to accuracy on unseen words (Table 7). All the differences are significant ($p < 0.05$ level) and have non-negligible effect size (Cohen’s $d > 0.2$). The results remain significant after applying the Bonferroni correction for multiple comparisons.

One may wonder if Eukalyptus has more difficult distinctions, or is more inconsistently annotated. However, it should be noted that the variation between splits is much larger for Eukalyptus than for the other two corpora. If we disregard testing on the blog part (although we still include it for training) the 4-fold macro average is more similar to the Talbanken-UD results, although still lower. However, the standard deviation (SD) is also still higher than for the other two corpora. The reason for this may be the distinctiveness of text types or genres of the Eukalyptus parts.

To check this, we also ran KB-Bert on randomized versions of the three corpora, where sentences are randomly assigned to folds. This means that the differences are evened out between folds and that the test data is more similar to the training data. The results are shown in Table 4. As we can see, the results between the three corpora are more similar than for the consecutive splits (with Eukalyptus even getting better results than Talbanken-UD). SD between folds is very low, except for Talbanken-UD. However, since the random assignment of sentences to splits makes tagging easier, all results reported in this paper, except for in Table 4, are based on the consecutive splits, not the random splits.

In Table 5 we look at sentence-level accuracy, that is the amount of sentences where all words have the correct tag. The pattern is the same as for the token-level results in Table 3 regarding which tagger performs the best, but the distance between Bert and the other taggers is even greater. However, the differences between folds are also greater.

4.3 Unseen words

Since training data can never contain all potential words or word-tag combinations, how well a tagger does on words previously unseen in the training data (OOV) is important, and often varies between different methods.

In Table 6 we show the numbers of unseen words, averaged over the five folds of each corpus. It is clear that the different folds for Talbanken-SBX and Talbanken-UD are quite similar, while there are larger differences between the folds of Eukalyptus. There, the Wikipedia part has the largest number of OOV word forms.

Table 7 shows tagging results for unseen words only. The only notable deviation from the general

	TB-SBX	TB-UD	Euk	Euk 4-fold
KB-Bert	97.71 (0.2)	97.28 (0.1)	96.64 (1.1)	97.14 (0.4)
Flair	97.31 (0.2)	96.79 (0.1)	95.88 (1.6)	96.63 (0.5)
Stanza	96.18 (0.3)	95.79 (0.1)	94.64 (1.7)	95.39 (0.8)
Marmot	95.62 (0.4)	94.94 (0.2)	93.75 (2.1)	94.72 (1.0)
Hunpos	93.58 (0.5)	92.85 (0.2)	91.31 (2.5)	92.33 (1.5)

Table 3: 5-fold macroaveraged accuracy for POS+MSD for all three corpora and all five taggers (standard deviation in parentheses). The final column shows a 4-fold macro average for Eukalyptus, excluding the blog part for testing.

TB-SBX	TB-UD	Euk
97.94 (0.05)	97.36 (0.11)	97.42 (0.04)

Table 4: 5-fold macroaveraged accuracy for POS+MSD for all three corpora using KB-Bert, where the data has been divided over the folds randomly (SD in parentheses).

	TB-SBX	TB-UD	Euk
KB-Bert	72.69 (4.5)	68.83 (3.4)	59.86 (5.2)
Flair	68.98 (4.9)	64.47 (2.7)	54.15 (5.8)
Stanza	60.10 (5.0)	57.55 (2.8)	46.27 (5.1)
Marmot	55.31 (4.6)	51.11 (2.6)	40.84 (5.2)
Hunpos	45.47 (4.4)	39.99 (2.1)	31.86 (5.4)

Table 5: 5-fold macroaveraged sentence-level accuracy for POS+MSD for all three corpora and all five taggers (SD in parentheses).

results is that Hunpos does equally well on unseen words for all three corpora. Given that Eukalyptus exhibits a large variation of unseen words, we examine the results per split. The results for the Blog fold are the worst (about 10 points lower POS+MSD-tagging accuracy on OOV tokens than the rest of the folds), while the number of OOV tokens in this fold is relatively low. This indicates that the unseen words in the blog data are difficult to tag given the context.

4.4 Difficult categories

If we look at the top-3 and bottom-3 POS tags, ranked by F1-score, for each fold and each tagger, we see that for Eukalyptus the worst tags are foreign words, interjections and proper nouns. Adverbs and adjectives appear among the bottom 3 once each (over all testfolds and all taggers). For Talbanken-SBX and Talbanken-UD the bottom is not as clear. The most frequent in the bottom 3

	TB-SBX	TB-UD	Euk
train	3377 (319)	3246 (257)	4368 (723)
train-dev	3076 (270)	2948 (242)	4065 (717)

Table 6: Average numbers of unseen words for the 5-fold test data sets (SD in parentheses). The train-dev data was used for training Hunpos and Marmot, while the train data only was used for KB-Bert, Flair, and Stanza.

	TB-SBX	TB-UD	Euk
KB-Bert	93.31 (0.4)	92.90 (0.4)	91.21 (3.2)
Flair	92.65 (0.6)	92.17 (0.4)	89.36 (3.8)
Stanza	88.65 (1.0)	88.49 (0.6)	85.33 (4.5)
Marmot	87.78 (0.9)	86.96 (0.7)	82.68 (5.8)
Hunpos	82.68 (3.5)	82.68 (3.2)	82.68 (12.6)

Table 7: 5-fold macroaveraged results for POS+MSD for previously unseen wordforms for all three corpora and all five taggers (SD in parentheses).

for Talbanken-SBX are foreign word, verb particle and interjection, while proper nouns, possessive wh-pronouns and wh-determiners appear a few times. Participles and ordinals appear only once. For Talbanken-UD symbols, subordinating conjunctions, interjections and proper nouns appear in the bottom 3 most frequently, while adverbs appear only twice.

Overall, this shows that interjections, foreign words, and proper nouns are difficult to predict correctly. This may not be surprising, since these categories generally apply to words with a high type count and there are no visible morphological cues. Foreign words additionally have a wide range of syntactic functions. Note that UD has a feature (MSD-tag) for foreign words, but not a POS-tag.

Another reason for these categories being difficult, at least in part, is that they are infrequent. Let us therefore explore categories with higher frequencies. Considering that there are generally around 20,000 tokens in the test sets, we can look at categories with more than 200 instances in the test data (ignoring categories with less than 1% of the test tokens each).

We see that for Eukalyptus, proper nouns, adjectives and adverbs are generally difficult, with foreign words, conjunctions and nouns also appearing in the bottom 3 at times. Hunpos seems to have more problems with nouns, however. Marmot has less difficulties with nouns, instead finding numerals slightly difficult. For Talbanken-SBX, participles are difficult, as well as proper nouns, adjectives and adverbs. Bert seems to also have problems with cardinals, but less with adverbs, while Marmot has less trouble with adjectives. For Talbanken-UD, the most difficult categories are proper nouns and subjunctions. Adverbs are also difficult for most taggers, although less so for Hunpos. Auxiliaries are a bit more difficult for Marmot and Hunpos, while numerals are bit more difficult for Bert, Flair and Stanza. Altogether, these differences can be exploited, for example in an ensemble approach (Section 4.6).

Looking at POS+MSD confusion matrices, we can see that one of the most frequent confusions (especially for both Talbankens) is that of singular and plural neuter indefinite nouns (in both directions). Indefinite singular and plural forms for Swedish neuter nouns ending in a consonant are syncretic (*barn* ‘child/children’, *hus* ‘house/houses’). The problem is exacerbated by the fact that at least in Talbanken-SBX, there are many contexts where the number of the noun cannot actually be inferred (both interpretations are possible). Such nouns, however, are not annotated as underspecified for number, but as either singular or plural, often inconsistently, which makes learning difficult. One example is shown in the example below. *Undantag* is tagged as plural according to the gold data, and as singular by KB-Bert, and both interpretations are possible.

- (1) *Undantag: periodiskt understöd eller*
 Exception(s): periodic support or
därmed jämförlig periodisk inkomst
 comparable periodic income

In Talbanken-UD, a frequent error concerns confusing verbs and auxiliaries. It seems to be that

the distinction between these two categories is not entirely consistently annotated in Talbanken-UD. In the following shortened examples, the gold data has different annotations for the verb *vara* ‘be’, although there is no clear difference between the two.

- (2) *Frågan är [AUX] om*
 The question is if
man med den konservativa grundsynen kan [...]
 one with the conservative basic view can
- (3) *Frågan är [VB] om*
 The question is if
synen på äktenskapet kan [...]
 the view of marriage can [...]

An issue particular to Eukalyptus is confusing symbols and punctuation. They are considered the same POS category, but two different MSD tags. This is not very surprising and seems to emerge from the amount of smileys in the blog fold. The result is a frequent mistagging of symbols as punctuation in the blog fold, and several cases of mistagging punctuation as symbols in the other folds, in particular in the novels. Many of the latter cases are quotation dashes, indicating a character’s speech. This method of marking direct speech is uncommon in the other types of texts.

4.5 What makes a tag difficult: quantitative analysis

We also perform a systematic statistical analysis of the factors which can potentially affect tagger performance. More specifically, we attempt to identify which properties make a tag difficult.

For every corpus, we concatenate all five test sets (i.e. microaverage across folds), and measure the following for every POS+MSD tag:

- the accuracy of every tagger on this tag;
- the frequency. The prediction is that frequent tags are easier to identify;
- type-token ratio (TTR) of tokens that have this tag. The prediction is that high TTR will make the tag more difficult to identify, cf. Section 4.4. TTR is strongly dependent on the sample size (less frequent tags are more likely to have higher TTR), but we judge that in this case, no correction is necessary;
- average “difficulty” of tokens that have this tag. This is done in two steps. First, we go through all tokens in the dataset, calculate the probability distribution of tags for every token and then the Shannon entropy of this

Predictor	Average (%)	SD	Significance
Frequency	0.003	0.0006	10/15
TTR	-85.2	6.4	15/15
Tag-by-token entropy	-27.0	7.4	15/15
Tag-by-ending entropy	6.8	3.1	10/15

Table 8: Summary of the regression models: average slope values and SD across all 15 models. Significance shows in how many of the models the predictor is significant at 0.05 level.

distribution. The entropy shows for every token how difficult it is to guess its tag and thus serves as a measure of “token difficulty”. At the second step, when analyzing a particular tag, we weigh the associated entropy by the relative frequency for every token that has this tag. We then sum the weighted values. The result (average conditional entropy) is meant to gauge how difficult on average the tokens that have the particular tag are;

- average “difficulty” of token endings (average entropy of tag conditioned on token ending). The procedure is exactly the same as for tokens, but instead of the whole token we are using its ending, which is typically the main grammatical marker in Swedish. For instance, *-er* can mark a present-tense verb or an indefinite plural noun. We are using the last two characters of the token as the ending (or the whole token if it’s shorter than two characters).

We fit a linear regression model with accuracy as the dependent variable (measured as percentage, i.e. on the 0–100 scale) and the four predictors described above as independent variables. We fit a separate model for every tagger and every corpus, i.e. 15 models in total. For all corpora, the collinearity of the predictors is very mild (the condition number varies from 8.2 to 9.5) and thus acceptable (Baayen, 2008, p. 181–182).

We summarize the results of the 15 models in Table 8. The results are very similar across corpora and folds for TTR and tag-by-token entropy, less so for frequency and tag-by-ending entropy. All models have high goodness-of-fit: the average multiple R^2 is 0.65, SD is 0.05.

In general, the first three predictions are borne out. On average, the increase in frequency by 1 token is expected to result in the increase in the tag accuracy by 0.003%. Frequency ranges from 1 to 11,000, which means that theoretically, the largest expected increase can be 33%.

The increase in tag-by-token entropy by 1 (note that this is a very large increase: entropy varies from 0 to 1.86 in our sample) is expected to decrease accuracy by 27%. The increase in TTR by 1 is expected to decrease accuracy by 85.2% (note that TTR cannot actually be larger than 1). TTR that is close to 0 is typical for tags that are assigned to a very small closed class of frequent tokens (e.g. punctuation marks). TTR of 1, on the contrary, can be achieved by tags that occur with (a few) very infrequent tokens (this is often a result of misannotation, or some very infrequent form or usage).

Surprisingly, the average conditional entropy of the tag given the ending goes directly against the prediction, yielding a positive effect (though small and not always significant). We cannot explain this effect. Our best guess is that high tag-by-ending entropy is correlated with some other properties that facilitate accurate tagging.

4.6 Ensemble

We tested whether combining the output of the five taggers may yield improved performance. In theory, it should be possible, since the proportion of cases where *at least one of the taggers* outputs a correct tag is higher than the accuracy of any individual tagger (see Table 9, row “Ceiling”).

We tried simple voting and a naive Bayes classifier (as implemented in the NBayes Ruby gem¹²). In both methods, the taggers are ordered by performance in descending order. In simple voting, each tagger gets one vote. In case of a tie, the vote that has come first wins. The naive Bayes classifier has to be trained. For that, we split the test set in each fold of each corpus into a training set (75%) and a test set (25%). What the classifier learns is how to match the input string (the token and the tags proposed by each tagger) with the label (which tagger makes the correct guess). If several taggers make a correct guess, the first one of those is chosen. If

¹²<https://github.com/oasic/nbayes>

Method	TB-SBX	TB-UD	Euk
Ceiling	99.16 (0.1)	98.78 (0.4)	98.26 (1.5)
KB-Bert	97.65 (0.3)	97.35 (0.6)	96.72 (1.6)
Voting	97.50 (0.1)	97.12 (1.0)	96.38 (2.2)
Bayes	97.65 (0.2)	97.41 (0.5)	96.75 (1.6)
Voting-fast	96.96 (0.3)	96.69 (0.7)	95.91 (1.8)
Bayes-fast	97.67 (0.3)	97.37 (0.6)	96.76 (1.7)

Table 9: Results of ensemble methods with comparison to the potential ceiling (at least one of the taggers guessed right) and the best single tagger (macroaveraged accuracy across all folds, SD in parentheses).

no taggers make a correct guess, KB-Bert is chosen by default. Changing this method (e.g. using only the tags as the input string) leads to slightly worse performance. Both voting and the classifier are then tested on the test set. Since Stanza and Flair are slow at training time, we also try a combination of the “fast” taggers: KB-Bert, Marmot and Hunpos.

The results are summarized in Table 9. Simple voting always performs worse than the best single tagger, but naive Bayes performs slightly better. For Talbanken-SBX and Eukalyptus, the best performance is achieved when the classifier is trained on the output of fast taggers only, while for Talbanken-UD the full training set yields better results. All differences are, however, very small. The difference between KB-Bert and Bayes is not significant ($t(14) = -1.1, p = 0.28, d = -0.03$), nor is the one between KB-Bert and Bayes-fast ($t(14) = -1.6, p = 0.12, d = -0.03$), no correction for multiple comparisons.

A possible avenue for future research would be to use other recently developed ensemble methods, as for instance Bohnet et al. (2018); Stoeckel et al. (2020).

5 Conclusions

We applied five taggers to three important Swedish corpora. The corpora are of comparable size and have different tagsets. Two of them consist of virtually the same texts, but are not entirely parallel.

We show that the three neural taggers outperform the two pre-neural (HMM and CRF) ones when it comes to tagging quality, but are significantly slower. KB-Bert, however, while always yielding the highest accuracy, is also the fastest of the neural taggers, and its speed on GPU is comparable with that of the pre-neural taggers.

Token-level accuracy of KB-Bert (97.2 on average across corpora) is very high, and is decent

also for OOV tokens (92.5). If we apply sentence-level accuracy, a less forgiving measure (Manning, 2011), we can see that there is actually much room for improvement (67.1).

The success of the taggers depends to a large extent on the additional data (embeddings, morphological dictionaries) that they receive as input, of which token embeddings (a.k.a. contextualized or dynamic) seem to be the most powerful ones. It is reasonable to assume that it is also important on which corpus the embeddings were trained. The size of this corpora is comparable for all neural taggers, but KB-Bert’s is likely to be the most balanced one.

The results vary across corpora/tagsets. If we use consecutive splits, TalbankenSBX always has the highest annotation accuracy and Eukalyptus the lowest one. The reason for that is that the two Talbankens are more homogeneous (contain only professional prose texts), while Eukalyptus contains texts from five different domains, one of which (blogs) is notoriously difficult. The reason for TalbankenSBX yielding better results than TalbankenUD is probably the less fine-grained tagset, but possibly also more consistent annotation. If, however, we use random splits, the accuracy for Eukalyptus goes up, surpassing the one for TalbankenUD.

Manual error analysis suggests that a high type count, absence of morphological cues, a wide range of syntactic functions, and low frequency make tags more difficult. Inconsistent annotation (which is very difficult to avoid in borderline cases) also seems to play an important role. We also perform a statistical analysis of the factors that can potentially affect how difficult the POS+MSD tags are. The regression model shows that type-token ratio within tag and average “difficulty” of tokens within tag (measured as entropy of guessing the tag given the token) have con-

sistently significant and very strong negative effects on the accuracy. Tag frequency has a positive (though not always significant) effect. Surprisingly, so does the average “difficulty” of token endings within tag (though the effect is small and not always significant). The results of the statistical analysis partly support the predictions done on the basis of the manual one. In general, this is a promising research avenue which deserves more systematic attention.

Finally, we test whether the tagger outputs can be combined using ensemble methods, since in theory, there clearly is a potential for that. In practice, it turns out that using a naive Bayes classifier it is possible to achieve a very small improvement over the best-performing tagger, but the difference is not statistically significant.

The data and scripts that are necessary to reproduce the regression analysis and the ensemble methods are available as supplementary materials¹³.

Acknowledgments

This work has been funded by Nationella Språkbanken – jointly funded by its 10 partner institutions and the Swedish Research Council (2018–2024; dnr 2017-00626). We would like to thank Gerlof Bouma, Simon Hengchen and Peter Ljunglöf for valuable comments on earlier versions of this paper.

References

- Yvonne Adesam and Gerlof Bouma. 2019. The Koala part-of-speech tagset. *Northern European Journal of Language Technology*, 6:5–41.
- Yvonne Adesam, Gerlof Bouma, and Richard Johansson. 2015. Defining the Eukalyptus forest – the Koala treebank of Swedish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania*. Edited by Beáta Megyesi, pages 1–9.
- Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.
- Harald Baayen. 2008. *Analyzing Linguistic Data: A Practical Introduction to Statistics using R*. Cambridge University Press.
- Bernd Bohnet, Ryan McDonald, Gonçalo Simões, Daniel Andor, Emily Pitler, and Joshua Maynez. 2018. Morphosyntactic tagging with a meta-BiLSTM model over context sensitive token encodings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2642–2652, Melbourne, Australia. Association for Computational Linguistics.
- Lars Borin, Markus Forsberg, Martin Hammarstedt, Dan Rosén, Roland Schäfer, and Anne Schumacher. 2016. Sparv: Språkbanken’s corpus annotation pipeline infrastructure. In *Swedish Language Technology Conference (SLTC)*. Umeå University.
- Lars Borin, Markus Forsberg, and Lennart Lönngrén. 2013. SALDO: a touch of yin to WordNet’s yang. *Language resources and evaluation*, 47(4):1191–1211.
- Lars Borin, Markus Forsberg, and Johan Roxendal. 2012. Korp – the corpus infrastructure of språkbanken. In *Proceedings of LREC 2012. Istanbul: ELRA*, page 474–478.
- Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. 1992. The linguistic annotation system of the Stockholm-Umeå corpus project - description and guidelines. Technical Report 33, Department of Linguistics, Umeå University.
- Per Fallgren, Jesper Segeblad, and Marco Kuhlmann. 2016. Towards a standard dataset of Swedish word vectors. In *Sixth Swedish Language Technology Conference (SLTC), Umeå 17-18 nov 2016*.
- Kyle Gorman and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791, Florence, Italy. Association for Computational Linguistics.
- Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. Hunpos: An open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL ’07*, page 209–212, USA. Association for Computational Linguistics.
- Martin Malmsten, Love Börjesson, and Chris Haf-fenden. 2020. Playing with words at the National Library of Sweden – making a Swedish BERT.
- Christopher D Manning. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *International conference on intelligent text processing and computational linguistics*, pages 171–189. Springer.

¹³<https://github.com/AleksandrsBerdicevskis/Swetagging2021>

- Beáta Megyesi. 2009. The open source tagger HunPoS for Swedish. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, pages 239–241, Odense, Denmark. Northern European Association for Language Technology (NEALT).
- Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA. Association for Computational Linguistics.
- Joakim Nivre. 2014. Universal Dependencies for Swedish. In Swedish Language Technology Conference (SLTC). Uppsala University.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA).
- Joakim Nivre and Beata Megyesi. 2007. Bootstrapping a Swedish treebank using cross-corpus harmonization and annotation projection. In *Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories*, pages 97–102.
- Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 1392–1395. European Language Resources Association (ELRA).
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.
- Manuel Stoeckel, Alexander Henlein, Wahed Hemati, and Alexander Mehler. 2020. Voting for POS tagging of Latin texts: Using the flair of FLAIR to better ensemble classifiers by example of Latin. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 130–135, Marseille, France. European Language Resources Association (ELRA).
- Anders Søgaard, Sebastian Ebert, Joost Bastings, and Katja Filippova. 2020. We need to talk about random splits.
- Ulf Teleman. 1974. *Manual för grammatisk beskrivning av talad och skriven svenska*. Studentlitteratur, Lund.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Misišilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Drohanova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkor-eit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

De-identification of Privacy-related Entities in Job Postings

Kristian Nørgaard Jensen,[◇] Mike Zhang[◇] and Barbara Plank

Department of Computer Science
ITU Copenhagen, Denmark

krnj@itu.dk, mikz@itu.dk, bplank@itu.dk

Abstract

De-identification is the task of detecting privacy-related entities in text, such as person names, emails and contact data. It has been well-studied within the medical domain. The need for de-identification technology is increasing, as privacy-preserving data handling is in high demand in many domains. In this paper, we focus on job postings. We present JOBSTACK, a new corpus for de-identification of personal data in job vacancies on Stackoverflow. We introduce baselines, comparing Long-Short Term Memory (LSTM) and Transformer models. To improve upon these baselines, we experiment with contextualized embeddings and distantly related auxiliary data via multi-task learning. Our results show that auxiliary data improves de-identification performance. Surprisingly, vanilla BERT turned out to be more effective than a BERT model trained on other portions of Stackoverflow.

1 Introduction

It is becoming increasingly important to anonymize privacy-related information in text, such as person names and contact details. The task of de-identification is concerned with detecting and anonymizing such information. Traditionally, this problem has been studied in the medical domain by e.g., Szarvas et al. (2007); Friedrich et al. (2019); Trienes et al. (2020) to anonymize (or pseudo-anonymize) person-identifiable information in electronic health records (EHR). With new privacy-regulations (Section 2) de-identification is becoming more important for broader types of text. For example, a company or public institution might seek to

de-identify documents before sharing them. On another line, de-identification can benefit society and technology at scale. Particularly auto-regressive models trained on massive text collections pose a potential risk for exposing private or sensitive information (Carlini et al., 2019, 2020), and de-identification can be one way to address this.

In this paper, we analyze how effectively sequence labeling models are in identifying privacy-related entities in job posts. To the best of our knowledge, we are the first study that investigates de-identification methods applied to job vacancies. In particular, we examine: How do Transformer-based models compare to LSTM-based models on this task (**RQ1**)? How does BERT compare to BERT_{Overflow} (Tabassum et al., 2020) (**RQ2**)? To what extent can we use existing medical de-identification data and Named Entity Recognition (NER) data to improve de-identification performance (**RQ3**)? To answer these questions, we put forth a new corpus, JOBSTACK, annotated with around 22,000 sentences in English job postings from Stackoverflow for person names, contact details, locations, and information about the profession of the job post itself.

Contributions We present JOBSTACK, the first job postings dataset with professional and personal entity annotations from Stackoverflow. Our experiments on entity de-identification with neural methods show that Transformers outperform bi-LSTMs, but surprisingly a BERT variant trained on another portion of Stackoverflow is less effective. We find auxiliary tasks from both news and the medical domain to help boost performance.

2 Related Work

2.1 De-identification in the Medical Domain

De-identification has mostly been investigated in the medical domain (e.g., Szarvas et al. (2007);

[◇]The authors contributed equally to this work.

Meystre et al. (2010); Liu et al. (2015); Jiang et al. (2017); Friedrich et al. (2019); Trienes et al. (2020)) to ensure the privacy of a patient in the analysis of their medical health records. Apart from an ethical standpoint, it is also a legal requirement imposed by multiple legislations such as the US Health Insurance Portability and Accountability Act (HIPAA) (Act, 1996) and the European General Data Protection Regulation (GDPR) (Regulation, 2016).

Many prior works in the medical domain used the I2B2/UTHealth dataset (Stubbs and Uzuner, 2015) to evaluate de-identification. The dataset consists of clinical narratives, which are free-form medical texts written as a first person account by a clinician. Each of the documents describes a certain event, consultation or hospitalization. All of the texts have been annotated with a set of Protected Health Information (PHI) tags (e.g. name, profession, location, age, date, contact, IDs) and subsequently replaced by realistic surrogates. The dataset was originally developed for use in a shared task for automated de-identification systems. Systems tend to perform very well on this set, in the shared task three out of ten systems achieved F1 scores above 90 (Stubbs et al., 2015). More recently, systems reach over 98 F1 with neural models (Dernoncourt et al., 2017; Liu et al., 2017; Khin et al., 2018; Trienes et al., 2020; Johnson et al., 2020). We took I2B2 as inspiration for annotation of JOBSTACK.

Past methods for de-identification in the medical domain can be categorised in three categories. (1) Rule-based approaches, (2) traditional machine learning (ML)-based systems (e.g., feature-based Conditional Random Fields (CRFs) (Lafferty et al., 2001), ensemble combining CRF and rules, data augmentation, clustering), and (3) neural-based approaches.

Rule-based First, Gupta et al. (2004) made use of a set of rules, dictionaries, and fuzzy string matching to identify protected health information (PHI). In a similar fashion, Neamatullah et al. (2008) used lexical look-up tables, regular expressions, and heuristics to find instances of PHI.

Traditional ML Second, classical ML approaches employ feature-based CRFs (Aberdeen et al., 2010; He et al., 2015). Moreover, earlier work showed the use of CRFs in an ensemble with rules (Stubbs et al., 2015). Other ML approaches

include data augmentation by McMurry et al. (2013), where they added public medical texts to properly distinguish common medical words and phrases from PHI and trained decision trees on the augmented data.

Neural methods Third, regarding neural methods, Dernoncourt et al. (2017) were the first to use Bi-LSTMs, which they used in combination with character-level embeddings. Similarly, Khin et al. (2018) performed de-identification by using a Bi-LSTM-CRF architecture with ELMo embeddings (Peters et al., 2018). Liu et al. (2017) used four individual methods (CRF-based, Bi-LSTM, Bi-LSTM with features, and rule-based methods) for de-identification, and used an ensemble learning method to combine all PHI instances predicted by the three methods. Trienes et al. (2020) opted for a Bi-LSTM-CRF as well, but applied it with contextual string embeddings (Akbik et al., 2018). Most recently, Johnson et al. (2020) fine-tuned BERT_{base} and BERT_{large} (Devlin et al., 2019) for de-identification. Next to “vanilla” BERT, they experiment with fine-tuning different domain specific pre-trained language models, such as SciBERT (Beltagy et al., 2019) and BioBERT (Lee et al., 2020). They achieve state-of-the art performance in de-identification on the I2B2 dataset with the fine-tuned BERT_{large} model. From a different perspective, the approach of Friedrich et al. (2019) is based on adversarial learning, which automatically pseudo-anonymizes EHRs.

2.2 De-identification in other Domains

Data protection in general however is not only limited to the medical domain. Even though work outside the clinical domain is rare, personal and sensitive data is in abundance in all kinds of data. For example, Eder et al. (2019) pseudonymised German emails. Bevendorff et al. (2020) published a large preprocessed email corpus, where only the email addresses themselves were anonymized. Apart from emails, several works went into de-identification of SMS messages (Treurniet et al., 2012; Patel et al., 2013; Chen and Kan, 2013) in Dutch, French, English and Mandarin respectively. Both Treurniet et al. (2012); Chen and Kan (2013) conducted the same strategy and automatically anonymized all occurrences of dates, times, decimal amounts, and numbers with more than one digit (telephone numbers, bank accounts, et cetera), email addresses, URLs, and IP ad-

	Train	Dev	Test	Total
Time	June – August 2020	September 2020		-
# Documents	313	41	41	395
# Sentences	18,055	2082	2092	22,219
# Tokens	195,425	22,049	21,579	239,053
# Entities	4,057	462	426	5,154
avg. # sentences	57.68	50.78	51.02	53.16
avg. tokens / sent.	10.82	10.59	10.32	10.78
avg. entities / sent.	0.22	0.22	0.20	0.21
density	14.73	14.31	14.58	14.54
Organization	1803	215	208	2226
Location	1511	157	142	1810
Profession	558	63	64	685
Contact	99	10	7	116
Name	86	17	5	108

Table 1: Statistics of our JOBSTACK dataset.

resses. All sensitive information was replaced with a placeholder. Patel et al. (2013) introduced a system to anonymize SMS messages by using dictionaries. It uses a dictionary of first names and anti-dictionaries (of ordinary language and of some forms of SMS writing) to identify the words that require anonymization.

In our work, we study de-identification for names, contact information, addresses, and professions, as further described in Section 3.

3 JOBSTACK Dataset

In this section, we describe the JOBSTACK dataset. There are two basic approaches to remove privacy-bearing data from the job postings. First, anonymization identifies instances of personal data (e.g. names, email addresses, phone numbers) and replaces these strings by some placeholder (e.g. {name}, {email}, {phone}). The second approach, pseudonymisation preserves the information of personal data by replacing these privacy-bearing strings with randomly chosen alternative strings from the same privacy type (e.g. replacing a name with “John Doe”). The term de-identification subsumes both anonymization and pseudonymisation. In this work, we focus on anonymization.¹

Eder et al. (2019) argues that the anonymization approach might be appropriate to eliminate privacy-bearing data in the medical domain, but

¹Meystre (2015) notes that de-identification means removing or replacing personal identifiers to make it difficult to reestablish a link between the individual and his or her data, but it does not make this link impossible.

would be inappropriate for most Natural Language Processing (NLP) applications since crucial discriminative information and contextual clues will be erased by anonymization.

If we shift towards pseudonymisation, we argue that there is still the possibility to resurface the original personal data. Henceforth, our goal is to anonymize job postings to the extent that one would not be able to easily identify a company from the job posting. However, as job postings are public, we are aware that it would be simple to find the original company that posted it with a search engine. Nevertheless, we abide to the GDPR compliance which requires us to protect the personal data and privacy of EU citizens for transactions that occur within EU member states (Regulation, 2016). In job postings this would be the names of employees, and their corresponding contact information.²

Over a period of time, we scraped 2,755 job postings from Stackoverflow and selected 395 documents to annotate, the subset ranges from June 2020 to September 2020. We manually annotated the job postings with the following five entities: Organization, Location, Contact, Name, and Profession.

To make the task as realistic as possible, we kept all sentences in the documents. The statistics pro-

²https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/application-regulation/do-data-protection-rules-apply-data-about-company_en

```

...
13. Job description:
14. [XXXOrganization] is a modern multi tenant, microservices based solution and Floor Planning is one major functional solution vertical of the [XXXOrganization] platform.

15. What you'll be doing:
16. As a [XXXProfession] for [XXXOrganization], you will be one of the founding members of our [XXXLocation] based floor planning development team.
17. You will be in charge for development of future floor planning capabilities on the [XXXOrganization] platform and be the software architect for the capability.
18. You will drive the team to improve the coding practices and boost performance.
19. You will also be a member of our [XXXOrganization] and have a major influence on feature roadmap and technologies we use.
...

```

Figure 1: Snippet of a job posting, full job posting can be found in Appendix A.

vided in the following therefore reflect the natural distribution of entities in the data. A snippet of an example job post can be seen in Figure 1, the full job posting can be found in Appendix A.

3.1 Statistics

Table 1 shows the statistics of our dataset. We split our data in 80% train, 10% development, and 10% test. Besides of a regular document-level random split, ours is further motivated based on time. The training set covers the job posts posted between June to August 2020 and the development- and test set are posted in September 2020. To split the text into sentences, we use the `sentence-splitter` library used for processing the Europarl corpus (Koehn, 2005). In the training set, we see that the average number of sentences is higher than in the development- and test set (6-7 more). We therefore also calculate the density of the entities, meaning the percentage of sentences with at least one entity. The table shows that 14.5% of the sentences in JOBSTACK contain at least one entity. Note that albeit having document boundaries, we treat the task of de-identification as a standard word-level sequence labeling task.

3.2 Annotation Schema

The aforementioned entity tags are based on the English I2B2/UTHealth corpus (Stubbs and Uzuner, 2015). The tags are more coarse-grained than the I2B2 tags. For example, we do not distinguish between zip code and city, but tag them with `Location`. We give a brief explanation of the tags.

Organization: This includes all companies and their legal entity mentioned in the job postings. The tag is not limited to the company that authored the job posting, but does also include men-

tions of stakeholders or any other company.

Location: This is the address of the company in the job posting. The location also refers to all other addresses, zip codes, cities, regions, and countries mentioned throughout the text. This is not limited to the company address, but should be used for all location names in the job posting, including abbreviations.

Contact: The label includes, URLs, email addresses and phone numbers. This could be, but is not limited to, contact info of an employee from the authoring company.

Name: This label covers names of people. This could be, but is not limited to, a person from the company, such as the contact person, CEO, or the manager. All names appearing in the job posting should be annotated no matter the relation to the job posting itself. Titles such as Dr. are not part of the annotation. Apart from people names in our domain, difficulties could arise with other type of names. An example would be project names, with which one could identify a company. In this work, we did not annotate such names.

Profession: This label covers the profession that is being searched for in the job posting or desired prior relevant jobs for the current profession. We do not annotate additional meta information such as gender (e.g. Software Engineer (f/m)). We also do not annotate mentions of colleague positions in neither singular or plural form. For example: “*As a Software Engineer, you are going to work with Security Engineers*”. Here we annotate Software Engineer as profession, but we do not annotate Security Engineers. While this may sound straightforward, however, there are difficulties in regards to annotating professions. A job posting is free text, meaning that one can write anything they prefer to make the job posting as clear as possible (e.g., *Software Engineer (at a unicorn start-*

up based in [..]). The opposite is also possible, when they are looking for one applicant to fill in one of multiple positions. For example, “*We are looking for an applicant to fill in the position of DevOps/Software Engineer*”. From our interpretation, they either want a “DevOps Engineer” or a “Software Engineer”. We decided to annotate the full string of characters “DevOps/Software Engineer” as a profession.

3.3 Annotation Quality

	Token	Entity	Unlabeled
A1 – A2	0.889	0.767	0.892
A1 – A3	0.898	0.782	0.904
A2 – A3	0.917	0.823	0.920
Fleiss’ κ	0.902	0.800	0.906

Table 2: Inter-annotator agreement of the annotators. We show agreement over pairs with Cohen’s κ and all annotators with Fleiss’ κ .

To evaluate our annotation guidelines, a sample of the data was annotated by three annotators, one with a background in Linguistics (A1) and two with a background in Computer Science (A2, A3). We used an open source text annotation tool named `DOCCANO` (Nakayama et al., 2018). There are around 1,500 overlapping sentences that we calculated agreement on. The annotations were compared using Cohen’s κ (Fleiss and Cohen, 1973) between pairs of annotators, and Fleiss’ κ (Fleiss, 1971), which generalises Cohen’s κ to more than two concurrent annotations. Table 2 shows three levels of κ calculations, we follow Balasuriya et al. (2009)’s approach of calculating agreement in NER. (1) `Token` is calculated on the token level, comparing the agreement of annotators on each token (including non-entities) in the annotated dataset. (2) `Entity` is calculated on the agreement between named entities alone, excluding agreement in cases where all annotators agreed that a token was not a named-entity. (3) `Unlabeled` refers to the agreement between annotators on the exact span match over the surface string, regardless of the type of named entity (i.e., we only check the position of tag without regarding the type of the named entity). Landis and Koch (1977) state that a κ value greater than 0.81 indicates almost perfect agreement. Given this, all annotators are in strong agreement.

After this annotation quality estimation, we finalized the guidelines. They formed the basis for the professional linguist annotator, who annotated and finalized the entire final `JOBSTACK` dataset.

4 Methods

For entity de-identification we use a classic Named Entity Recognition (NER) approach using a Bi-LSTM with a CRF layer. On top of this we evaluate the performance of Transformer-based models with two different pre-trained BERT variants. Furthermore, we evaluate the helpfulness of auxiliary tasks, both using data close to our domain, such as de-identification of medical notes, and more general NER, which covers only a subset of the entities. Further details on the data are given in Section 4.3.

4.1 Models

Firstly, we test a Bi-LSTM sequence tagger (Bilty) (Plank et al., 2016), both with and without a CRF layer. The architecture is similar to the widely used models in previous works. For example, preliminary results of Bilty versus Trienes et al. (2020) show accuracy almost identical to each other: 99.62% versus 99.76%. Next we test a Transformer based model, namely the `MaChAmp` (van der Goot et al., 2021) toolkit. Current research shows good results for NER using a Transformer model without a CRF layer (Martin et al., 2020), hence we tested `MaChAmp` both with and without a CRF layer for predictions. For both models, we use their default parameters.

4.2 Embeddings

For embeddings, we tested with no pre-trained embeddings, pre-trained GloVe (Pennington et al., 2014) embeddings, and Transformer-based pre-trained embeddings. For Transformer-based embeddings we focused our attention on two BERT models, `BERTbase` (Devlin et al., 2019) and `BERTOverflow` (Tabassum et al., 2020). When using the Transformer-based embeddings with the Bi-LSTM, the embeddings were fixed and did not get updated during training.

Using the `MaChAmp` (van der Goot et al., 2021) toolkit, we fine-tune the BERT variant with a Transformer encoder. For the Bi-LSTM sequence tagger, we first derive BERT representations as input to the tagger. The tagger further uses word

Model	F1 Score	Precision	Recall
Bilty	71.76 \pm 2.57	79.00 \pm 1.10	65.80 \pm 3.72
Bilty + CRF	75.15 \pm 0.66	84.09 \pm 1.90	67.96 \pm 0.81
Bilty + Glove 50d	72.53 \pm 0.83	79.21 \pm 2.19	67.03 \pm 2.76
Bilty + Glove 50d + CRF	72.74 \pm 2.23	82.93 \pm 0.87	64.93 \pm 3.93
Bilty + BERT _{base}	77.99 \pm 0.91	83.70 \pm 0.58	73.01 \pm 1.34
Bilty + BERT _{base} + CRF	80.09 \pm 0.60	88.23 \pm 0.87	73.30 \pm 1.47
Bilty + BERT _{Overflow}	52.01 \pm 3.15	70.86 \pm 0.68	41.27 \pm 4.19
Bilty + BERT _{Overflow} + CRF	53.08 \pm 2.88	77.79 \pm 1.20	40.33 \pm 2.98
MaChAmp + BERT _{base}	85.70 \pm 0.13	86.66 \pm 0.73	84.78 \pm 0.44
MaChAmp + BERT _{base} + CRF	86.27 \pm 0.31	86.40 \pm 0.62	86.15 \pm 0.00
MaChAmp + BERT _{Overflow}	65.84 \pm 0.48	70.88 \pm 0.17	61.47 \pm 0.81
MaChAmp + BERT _{Overflow} + CRF	69.35 \pm 0.96	77.27 \pm 3.68	63.06 \pm 2.11

Table 3: Results on the development set across three runs using our JOBSTACK dataset.

Model	Auxiliary tasks	F1 Score	Precision	Recall
Bilty + BERT _{base} + CRF	JOBSTACK + CoNLL	81.90 \pm 0.32	86.91 \pm 1.94	77.49 \pm 1.87
	JOBSTACK + I2B2	79.15 \pm 2.19	83.61 \pm 2.61	75.18 \pm 2.59
	JOBSTACK + CoNLL + I2B2	81.37 \pm 2.01	84.92 \pm 1.67	78.28 \pm 4.34
Bilty + BERT _{Overflow} + CRF	JOBSTACK + CoNLL	58.62 \pm 1.46	79.34 \pm 2.34	46.54 \pm 1.99
	JOBSTACK + I2B2	55.99 \pm 1.93	72.03 \pm 6.48	46.10 \pm 2.55
	JOBSTACK + CoNLL + I2B2	59.15 \pm 2.15	71.20 \pm 4.80	50.86 \pm 3.31
MaChAmp + BERT _{base} + CRF	JOBSTACK + CoNLL	87.20 \pm 0.34	87.24 \pm 1.94	87.23 \pm 1.24
	JOBSTACK + I2B2	86.64 \pm 0.53	88.44 \pm 0.84	84.92 \pm 0.44
	JOBSTACK + CoNLL + I2B2	86.06 \pm 0.66	86.13 \pm 0.50	86.00 \pm 0.87
MaChAmp + BERT _{Overflow} + CRF	JOBSTACK + CoNLL	70.62 \pm 0.64	75.65 \pm 1.41	66.24 \pm 0.98
	JOBSTACK + I2B2	73.88 \pm 0.16	80.26 \pm 1.32	68.47 \pm 1.03
	JOBSTACK + CoNLL + I2B2	73.29 \pm 0.22	77.66 \pm 0.82	69.41 \pm 0.89

Table 4: Performance of multi-task learning on the development set across three runs.

and character embeddings which are updated during model training.

The BERT_{Overflow} model is a transformer with the same architecture as BERT_{base}. It has been trained from scratch on a large corpus of text from the Q&A section of Stackoverflow, making it closer to our text domain than the “vanilla” BERT model. However, BERT_{Overflow} is not trained on the job postings portion of Stackoverflow.

4.3 Auxiliary tasks

Both the Bi-LSTM (Plank et al., 2016) and the MaChAmp (van der Goot et al., 2021) toolkit are capable of Multi Task Learning (MTL) (Caruana, 1997). We therefore, set up a number of experiments testing the impact of three different auxiliary tasks. The auxiliary tasks and their datasets are as follows:

- I2B2/UTHealth (Stubbs and Uzuner, 2015) - Medical de-identification;
- CoNLL 2003 (Sang and De Meulder, 2003) - News Named Entity Recognition;
- The combination of the above.

The data of the two tasks are similar to our dataset in two different ways. The I2B2 lies in a different text domain, namely medical notes, however, the label set of the task is close to our label set, as mentioned in Section 3.2. For CoNLL, we have a general corpus of named entities but fewer types (location, organization, person, and miscellaneous), but the text domain is presumably closer to our data. We test the impact of using both auxiliary tasks along with our own dataset.

Model	Auxiliary tasks	F1 Score	Precision	Recall
Bilty + BERT _{base} + CRF	JOBSTACK	78.99 ± 0.32	82.44 ± 0.95	75.90 ± 1.39
MaChAmp + BERT _{base} + CRF	JOBSTACK	79.91 ± 0.38	75.92 ± 0.39	84.35 ± 0.49
	JOBSTACK + CoNLL	81.27 ± 0.28	77.84 ± 1.19	85.06 ± 0.91
	JOBSTACK + I2B2	82.05 ± 0.80	80.30 ± 0.99	83.88 ± 0.67
	JOBSTACK + CoNLL + I2B2	81.47 ± 0.43	77.66 ± 0.58	85.68 ± 0.57

Table 5: Evaluation of the best performing models on the test set across three runs.

5 Evaluation

Here we will outline the results of the experiments described in Section 4. All results are mean scores across three different runs.³ The metrics are all calculated using the `conlleval` script⁴ from the original CoNLL-2000 shared task. Table 3 shows the results from training on JOBSTACK only, Table 4 shows the results of the MTL experiments described in Section 4.3. Both report results on the development set. Lastly, Table 5 shows the scores from evaluating selected best models as found on the development set, when tested on the final held-out test set.

Is a CRF layer necessary? In Table 3, as expected, adding the CRF for the Bi-LSTM clearly helps, and consistently improves precision and thereby F1 score. For the stronger BERT model the overall improvement is smaller and does not necessarily stem from higher precision. We note that on average across the three seed runs, MaChAmp with BERT_{base} and no CRF mistakenly adds an I-tag following an O-tag 8 times out of 426 gold entities. In contrast, the MaChAmp with BERT_{base} and CRF, makes no such mistake in any of its three seed runs. Earlier research, such as Souza et al. (2019) show that BERT models with a CRF layer improve or perform similarly to its simpler variants when comparing the overall F1 scores. Similarly, they note that in most cases it shows higher precision scores but lower recall, as in our results for the development set. However, interestingly, the precision drops during test for the Transformer-based model. As the overall F1 score increases slightly, we use the CRF layer in all subsequent experiments. The main take-away here is that both models benefit from an added CRF layer for the task, but the Transformer model

to a smaller degree.

LSTM versus Transformer Initially, LSTM networks dominated the de-identification field in the medical domain. Up until recently, large-scale pre-trained language models have been ubiquitous in NLP, although rarely used in this field. On both development and test results (Table 3, Table 5), we show that a Transformer-based model outperforms the LSTM-based approaches with non-contextualized and contextualized representations.

Poor performance with BERT_{Overflow} BERT_{base} is the best embedding method among all experiments using Bilty, with BERT_{Overflow} being the worst with a considerable margin. Being able to fine-tune BERT_{base} does give a good increase in performance overall. The same trend is apparent with fine-tuning BERT_{Overflow}, but it is not enough to catch up with BERT_{base}. We see that overall MaChAmp with BERT_{base} and CRF is the best model. However, Bilty with BERT_{base} and CRF does have the best precision.

We hypothesized the domain-specific BERT_{Overflow} representations would be beneficial for this task. Intuitively, BERT_{Overflow} would help with detecting profession entities. Profession entities contain specific skills related to the IT domain, such as *Python developer*, *Rust developer*, *Scrum master*. Although the corpus it is trained on is not one-to-one to our vacancy domain, we expected to see at most a slight performance drop. This is not the case, as the drop in performance turned out to be high. It is not fully clear to us why this is the case. It could be the Q&A data it is trained on consists of more informal dialogue than in job postings. In the future, we would like to compare these results to training a BERT model on job postings data.

Auxiliary data increases performance Looking at the results from the auxiliary experiments in Table 4 we see that all auxiliary sources are ben-

³We sampled three random seeds: 3477689, 4213916, 8749520 which are used for all experiments.

⁴<https://www.clips.uantwerpen.be/conll2000/chunking/output.html>

eficial, for both types of models. A closer look reveals that once again MaChAmp with BERT_{base} is the best performer across all three auxiliary tasks. Also, we see that Bilty with BERT_{base} has good precision, though not the best this time around. For a task like de-identification recall is preferable, thereby showing that fine-tuning BERT is better than the classic Bi-LSTM-CRF. Moreover, we see that BERT_{Overflow} is under-performing compared to BERT_{base}. However, BERT_{Overflow} is able to get a four point increase in F1 with I2B2 as auxiliary task in MaChAmp. For Bilty with BERT_{Overflow} we see a slightly greater gain with both CoNLL and I2B2 as auxiliary tasks. When comparing the auxiliary data sources to each other, we note that the closer text domain (CoNLL news) is more beneficial than the closer label set (I2B2) from a more distant medical text source. This is consistent for the strongest models.

In general, it can be challenging to train multi-task networks that outperform or even match their single-task counterparts (Alonso and Plank, 2017; Clark et al., 2019). Ruder (2017) mentions training on a large number of tasks is known to help regularize multi-task models. A related benefit of MTL is the transfer of learned “knowledge” between closely related tasks. In our case, it has been beneficial to add auxiliary tasks to improve our performance on both development and test compared to a single task setting. In particular, it seemed to have helped with pertaining a high recall score.

Performance on the test set Finally, we evaluate the best performing models on our held out test set. The best models are selected based on their performance on F1, precision, and recall. The results are seen in Table 5. Comparing the results to those seen in Table 3 and Table 4 it is clear to see that Bilty with BERT_{base} sees a smaller drop in F1 compared to that of MaChAmp with BERT_{base}. We do also see an increase in recall for Bilty compared to its performance on the development set. In general we see that recall for each model is staying quite stable without any significant drops. It is also interesting to see that, the internal ranking between MTL MaChAmp with BERT_{base} has changed, with JOBSTACK + I2B2 being the best performing model in terms of F1.

Per-entity Analysis In Table 6, we show a deeper analysis on the test set: the performance of the two different auxiliary tasks in a multi-

Entity		MaChAmp + JOBSTACK	
		+ CoNLL	+ I2B2
Organization (208)	F1	77.51 ± 0.81	78.34 ± 1.32
	P	73.73 ± 1.66	77.86 ± 1.60
	R	81.73 ± 0.96	78.85 ± 1.74
Location (142)	F1	86.88 ± 1.51	86.67 ± 1.80
	P	83.86 ± 1.82	83.47 ± 1.19
	R	90.14 ± 1.41	90.14 ± 2.54
Profession (64)	F1	80.20 ± 2.76	83.88 ± 0.90
	P	77.44 ± 3.82	82.42 ± 0.63
	R	83.33 ± 4.51	85.42 ± 1.80
Contact (7)	F1	87.91 ± 3.81	75.48 ± 4.30
	P	90.47 ± 8.25	71.03 ± 4.18
	R	85.71 ± 0.00	80.95 ± 8.24
Name (5)	F1	86.25 ± 8.08	85.86 ± 4.38
	P	76.39 ± 12.03	75.40 ± 6.87
	R	100.00 ± 0.00	100.00 ± 0.00

Table 6: Performance of the two different auxiliary tasks. Reported is the F1, Precision (P), and Recall (R) per entity. The number behind the entity name is the gold label instances in the test set.

task learning setting, namely CoNLL and I2B2. We hypothesized different performance gains with each auxiliary task. For I2B2, we expected Contact and Profession to do better than CoNLL, since I2B2 contains contact information entities (e.g., phone numbers, emails, et cetera) and professions of patients. Surprisingly, this is not the case for Contact, as CoNLL outperforms I2B2 on all three metrics. We do note however this result could be due to little instances of Contact and Name being present in the gold test set. Additionally, both named entities are predicted six to nine times by both models on all three runs on the test set. This could indicate the strong difference in performance. For Profession, it shows that I2B2 is beneficial for this particular named entity as expected. For the other three named entities, the performance is similar. As Location, Name, and Organization are in both datasets, we did not expect any difference in performance. The results confirm this intuition.

6 Conclusions

In this work we introduce JOBSTACK, a dataset for de-identification of English Stackoverflow job postings. Our implementation is publicly available.⁵ The dataset is freely available upon request.

We present neural baselines based on LSTM

⁵<https://github.com/kris927b/JobStack>

and Transformer models. Our experiments show the following: (1) Transformer-based models consistently outperform Bi-LSTM-CRF-based models that have been standard for de-identification in the medical domain (**RQ1**). (2) Stackoverflow-related BERT representations are not more effective than regular BERT representations on Stackoverflow job postings for de-identification (**RQ2**). (3) MTL experiments with BERT representations and related auxiliary data sources improve our de-identification results (**RQ3**); the auxiliary task trained on the closer text type was the most beneficial, yet results improved with both auxiliary data sources. This shows the benefit of using multi-task learning for de-identification in job vacancy data.

Acknowledgements

We thank the NLPnorth group for feedback on an earlier version of this paper. We would also like to thank the anonymous reviewers for their comments to improve this paper. Last, we also thank NVIDIA and the ITU High-performance Computing cluster for computing resources. This research is supported by the Independent Research Fund Denmark (DFF) grant 9131-00019B.

References

- John Aberdeen, Samuel Bayer, Reyhan Yeniterzi, Ben Wellner, Cheryl Clark, David Hanauer, Bradley Malin, and Lynette Hirschman. 2010. The mitre identification scrubber toolkit: design, training, and assessment. *International journal of medical informatics*, 79(12):849–859.
- Accountability Act. 1996. Health insurance portability and accountability act of 1996. *Public law*, 104:191.
- Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.
- Héctor Martínez Alonso and Barbara Plank. 2017. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 44–53.
- Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R Curran. 2009. Named entity recognition in wikipedia. In *Proceedings of the 2009 Workshop on The People’s Web Meets NLP: Collaboratively Constructed Semantic Resources (People’s Web)*, pages 10–18.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3606–3611.
- Janek Bevendorff, Khalid Al Khatib, Martin Potthast, and Benno Stein. 2020. Crawling and preprocessing mailing lists at scale for dialog analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1151–1158.
- Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 267–284.
- Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, et al. 2020. Extracting training data from large language models. *arXiv preprint arXiv:2012.07805*.
- Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.
- Tao Chen and Min-Yen Kan. 2013. Creating a live, public short message service corpus: the nus sms corpus. *Language Resources and Evaluation*, 47(2):299–335.
- Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D Manning, and Quoc V Le. 2019. Bam! born-again multi-task networks for natural language understanding. *arXiv preprint arXiv:1907.04829*.
- Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. 2017. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.
- Elisabeth Eder, Ulrike Krieg-Holz, and Udo Hahn. 2019. De-identification of emails: Pseudonymizing privacy-sensitive data in a german email corpus. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 259–269.
- Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

- Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619.
- Max Friedrich, Arne Köhn, Gregor Wiedemann, and Chris Biemann. 2019. Adversarial learning of privacy-preserving text representations for de-identification of medical records. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5829–5839, Florence, Italy. Association for Computational Linguistics.
- Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive Choice, Ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the Software Demonstrations of the 16th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Dilip Gupta, Melissa Saul, and John Gilbertson. 2004. Evaluation of a deidentification (de-id) software engine to share pathology reports and clinical documents for research. *American journal of clinical pathology*, 121(2):176–186.
- Bin He, Yi Guan, Jianyi Cheng, Keting Cen, and Wenlan Hua. 2015. Crfs based de-identification of medical records. *Journal of biomedical informatics*, 58:S39–S46.
- Zhipeng Jiang, Chao Zhao, Bin He, Yi Guan, and Jingchi Jiang. 2017. De-identification of medical records using conditional random fields and long short-term memory networks. *Journal of biomedical informatics*, 75:S43–S53.
- Alistair EW Johnson, Lucas Bulgarelli, and Tom J Pollard. 2020. Deidentification of free-text medical records using pre-trained bidirectional transformers. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 214–221.
- Kaung Khin, Philipp Burckhardt, and Rema Padman. 2018. A deep learning architecture for de-identification of patient notes: Implementation and evaluation. *arXiv preprint arXiv:1810.01570*.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.
- John D Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.
- J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.
- Zengjian Liu, Yangxin Chen, Buzhou Tang, Xiaolong Wang, Qingcai Chen, Haodi Li, Jingfeng Wang, Qiwen Deng, and Suisong Zhu. 2015. Automatic de-identification of electronic medical records using token-level and character-level conditional random fields. *Journal of biomedical informatics*, 58:S47–S52.
- Zengjian Liu, Buzhou Tang, Xiaolong Wang, and Qingcai Chen. 2017. De-identification of clinical notes via recurrent neural network and conditional random field. *Journal of biomedical informatics*, 75:S34–S42.
- Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.
- Andrew J McMurry, Britt Fitch, Guergana Savova, Isaac S Kohane, and Ben Y Reis. 2013. Improved de-identification of physician notes through integrative modeling of both public and private medical text. *BMC medical informatics and decision making*, 13(1):112.
- Stephane M Meystre. 2015. De-identification of unstructured clinical data for patient privacy protection. In *Medical Data Privacy Handbook*, pages 697–716. Springer.
- Stephane M Meystre, F Jeffrey Friedlin, Brett R South, Shuying Shen, and Matthew H Samore. 2010. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC medical research methodology*, 10(1):70.
- Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text annotation tool for human. Software available from <https://github.com/doccano/doccano>.
- Ishna Neamatullah, Margaret M Douglass, H Lehman Li-wei, Andrew Reisner, Mauricio Villarroel, William J Long, Peter Szolovits, George B Moody, Roger G Mark, and Gari D Clifford. 2008. Automated de-identification of free-text medical records. *BMC medical informatics and decision making*, 8(1):32.
- Namrata Patel, Pierre Accorsi, Diana Inkpen, Cédric Lopez, and Mathieu Roche. 2013. Approaches of anonymisation of an sms corpus. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 77–88. Springer.

- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.
- Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany. Association for Computational Linguistics.
- General Data Protection Regulation. 2016. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46. *Official Journal of the European Union (OJ)*, 59(1-88):294.
- Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.
- Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.
- Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2019. Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*.
- Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. 2015. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1. *Journal of biomedical informatics*, 58:S11–S19.
- Amber Stubbs and Özlem Uzuner. 2015. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus. *Journal of Biomedical Informatics*, 58:S20–S29.
- György Szarvas, Richárd Farkas, and Róbert Busa-Fekete. 2007. State-of-the-art anonymization of medical records using an iterative machine learning framework. *Journal of the American Medical Informatics Association*, 14(5):574–580.
- Jeniya Tabassum, Mounica Maddela, Wei Xu, and Alan Ritter. 2020. Code and named entity recognition in stackoverflow. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Maaske Treurniet, Orphee De Clercq, Henk van den Heuvel, and Nelleke Oostdijk. 2012. Collection of a corpus of dutch sms. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 2268–2273.
- J Trienes, D Trieschnigg, C Seifert, and D Hiemstra. 2020. Comparing rule-based, feature-based and deep neural methods for de-identification of dutch medical records. In *Eickhoff, C.(ed.), Health Search and Data Mining Workshop: Proceedings of the ACM WSDM 2020 Health Search and Data Mining Workshop co-located with the 13th ACM International WSDM Conference (WSDM 2020) Houston, Texas, USA, February 3, 2020*, pages 3–11. [SI]: CEUR.

A Example Job Posting

1. [XXXProfession]
2. [XXXOrganization]
3. <ADDRESS>, <ADDRESS>, [XXXLocation], - , [XXXLocation]
4. Date posted: 2020-08-13
5. Likes: 0, Dislikes: 0, Love: 0
6. Salary: SALARY
7. Job type: FULL_TIME
8. Experience level: Mid-Level, Senior, Lead
9. Industry: Big Data, Cloud-Based Solutions, Enterprise Software
10. Company size: 501-1
11. Company type: Private
12. Technologies: c#, typescript, cad, 2d, 3d
13. Job description:
14. [XXXOrganization] is a modern multi tenant, microservices based solution and Floor Planning is one major functional solution vertical of the [XXXOrganization] platform.
15. What you'll be doing:
16. As a [XXXProfession] for [XXXOrganization], you will be one of the founding members of our [XXXLocation] based floor planning development team.
17. You will be in charge for development of future floor planning capabilities on the [XXXOrganization] platform and be the software architect for the capability.
18. You will drive the team to improve the coding practices and boost performance.
19. You will also be a member of our [XXXOrganization] and have a major influence on feature roadmap and technologies we use.
20. What you'll bring to the table:
21. Solid software design and development skills and at least 5 year experience in the industry
22. Good understanding of CAD type of software in 2D and 3D worlds
23. Experience on rendering technologies, 2D/3D data models and data types
24. Hands-on experience in implementing CAD designers / drafting / drawing tools for on-line use
25. C#, C++, TypeScript or Angular/React knowledge
26. Strong ambition to deliver great quality software and to continuously improve the way we do development
27. Good spoken and written English
28. Ability to work on-site in our [XXXLocation] office, with flexible remote work possibilities
29. What we consider as an advantage:
30. Eagerness to find out and learn about the latest computer graphics technologies, and also to share your findings
31. Knowledge of OpenDesign components (Teigha)
32. What we offer you in return:
33. An international career and learning opportunities in a rapidly growing software company
34. A fun, ambitious, and committed team of smart people to work with
35. A respectful and professional, yet easy-going atmosphere where individual thinking is encouraged
36. Responsibilities in challenging projects from day one
37. A position where you can help retailers fight against food waste
38. Are you the one we're looking for?
39. Apply today and become a part of our [XXXOrganization] family!
40. You can apply by sending your cover letter and resume through the application form as soon as possible, but no later than 31st of August.
41. Please note that we will fill this position as soon as we've found the right person, so we recommend that you act quickly.
42. If you have questions, [XXXName] ([XXXContact]) from our Recruitment team is happy to answer them.
43. Also kindly note that we cannot process any applications through email.
44. Job benefits:
<cutoff>
53. Company description:
54. [XXXOrganization] is a fast-growing software company developing products that help retail companies plan and operate more efficiently.
55. By accurately forecasting consumption of goods, we reduce inventory costs, increase availability and cut waste.
56. Helping retailers eliminate food spoilage and reduce fleet emissions from transportation has a significant environmental impact as well!

Creating and Evaluating a Synthetic Norwegian Clinical Corpus for De-Identification

Synnøve Bråten

Department of Computer
and Systems Sciences
Stockholm University
Kista, Sweden

synnovebr@hotmail.com

Wilhelm Wie

Department of Computer
and Systems Sciences
Stockholm University
Kista, Sweden

w.wie@gmx.com

Hercules Dalianis

Department of Computer
and Systems Sciences
Stockholm University
Kista, Sweden

hercules@dsv.su.se

Abstract

Building tools to remove sensitive information such as personal names, addresses, and telephone numbers - so called Protected Health Information (PHI) - from clinical free text is an important task to make clinical texts available for research. These de-identification tools must be assessed regarding their quality in the form of the measurements precision and recall. To assess such tools, gold standards - annotated clinical text - must be available. Such gold standards exist for larger languages. For Norwegian, however, there are no such resources. Therefore, an already existing Norwegian synthetic clinical corpus, *NorSynthClinical*, has been extended with PHIs and annotated by two annotators, obtaining an inter-annotator agreement of 0.94 F_1 -measure. In total, the corpus has 409 annotated PHI instances and is called *NorSynthClinical PHI*. A de-identification hybrid tool (machine learning and rule-based methods) for Norwegian was developed and trained with open available resources, and obtained an overall F_1 -measure of 0.73 and a recall of 0.62, when evaluated using *NorSynthClinical PHI*. *NorSynthClinical PHI* is made open and available at Github to be used by the research community.

1 Introduction

The data contained within Electronic Health Records (EHRs) are of significant value to medical researchers and for administrative purposes, but privacy and patient confidentiality legislation restricts access. However, de-identification of such data - removing the Protected Health Information (PHI) within - allows it to be shared between

researchers (El Emam et al., 2009). This process can be done manually; however, manual de-identification has proven to be inefficient with regards to cost, time and quality (Dernoncourt et al., 2017).

Tools for automatic de-identification of clinical data have been studied extensively. However, most of the published research is concerned with structured records and not clinical free-text, and few de-identification tools are made publicly available (Neamatullah et al., 2008). Furthermore, most research focus on English and other languages with many native speakers. Despite the fact that the Norwegian language has comparatively few native speakers¹, hospitals and organisations like the Cancer Registry of Norway are in possession of comprehensive collections of clinical data. Enabling research on this valuable and unique information could reveal new discoveries and would be of great importance for the future health care.

To ensure that de-identification applications can successfully de-identify clinical texts, they must be evaluated in a quantitative manner (Dalianis, 2018). For this purpose, verified, annotated corpora are used to test and score the applications (Pustejovsky and Stubbs, 2012). These corpora are referred to as gold standards (or reference standards), and are typically made by domain experts or linguists - following specific guidelines. A gold standard does not need to contain real PHI, and it can be developed using synthetic data. Consequently, a gold standard developed with synthetic data can be made publicly available.

This study describes the efforts of creating and evaluating the first publicly available gold standard for de-identification of Norwegian Bokmål² clinical text, describing and discussing the devel-

¹Norwegian has approximately 4,320,000 native speakers, (Rehm and Uszkoreit, 2012)

²Norwegian Bokmål - One of the two official written variants of Norwegian.

opment and evaluation of the gold standard.

2 Related research

Marimon et al. (2019) created a gold standard corpus of Spanish synthetic clinical text. The corpus is called *Spanish Medical Document Anonymization (MEDDOCAN)* and consists of 250 clinical cases manually enriched with PHI phrases. The gold standard was applied in a community challenge track in order to evaluate the performance of de-identification tools focusing on the Spanish language. 63 systems were evaluated and 61 received an F_1 -measure score above 0.70, and the highest score was 0.97. As the gold standard seems to have served its purpose, Marimon et al. (2019) provides a good example of how to solve data sparsity problems.

The lack of publicly available clinical text in Norwegian places limitations on the development of gold standards and tools for de-identification of Norwegian clinical text. Recently, there have been developments of open datasets for Named Entity Recognition (NER) of the Norwegian language, most notably *NorSynthClinical* (Rama et al., 2018) and *NorNE* (Jørgensen et al., 2020). *NorSynthClinical* is a small dataset of synthetic clinical text, focusing on family history information (further described in Section 3) (Rama et al., 2018). While the development of *NorNE* resulted in a sizeable dataset with approximately 300,000 tokens for each written variant of Norwegian and a rich entity set, most PHI entity types are missing (Jørgensen et al., 2020).

Only a few attempts aiming at developing de-identification tools focusing on the Norwegian language have previously been made. One of these was conducted by Bjurstrøm and Singh (2013). They tackled de-identification of Norwegian free text clinical notes for their master’s thesis project, employing a combination of pattern recognition and simplistic statistical methods, reporting an F_1 -measure of 0.72. Furthermore, they developed a reference in order to evaluate their developed tool, consisting of 225 records manually annotated and de-identified. It was, however, not evaluated further or made publicly available (Bjurstrøm and Singh, 2013).

As previously mentioned, most of the existing tools and gold standards for de-identification of clinical text are written in, and for, the English language (Dalianis, 2018). One of the most well-

known gold standards is the *Multiparameter Intelligent Monitoring in Intensive Care (MIMIC II)* corpus (Saeed et al., 2002).

In Sweden, the development of both de-identification tools and gold standards has come further than in Norway. In 2008, a group of Swedish researchers developed a gold standard corpus for de-identification of Swedish clinical text (Velupillai et al., 2009). The researchers manually annotated and de-identified 100 electronic patient records (EPRs) deriving from five different clinics (*Neurology, Orthopaedia, Infection, Dental Surgery and Nutrition*) at Karolinska University Hospital. The gold standard consists of unstructured text (around 174,000 tokens in total) and is known as the *Stockholm EPR PHI* corpus. It has 4,700 annotated instances distributed over 8 PHI-classes. It has been further developed to *Stockholm EPR PHI Pseudo* corpus, which contains only surrogate names, addresses, phone numbers, etc., and is partly available for research (Dalianis, 2019).

3 Data

3.1 NorSynthClinical

A corpus of Norwegian synthetic clinical text, the *NorSynthClinical* corpus³, formed the basis of the created gold standard. *NorSynthClinical* is considered the first publicly available resource of Norwegian clinical text (Rama et al., 2018). It is written by one clinician with large experience with clinical work and genetic cardiology. The corpus describes patients’ family history relating to cases of cardiac disease, and according to Rama et al. (2018), it consists of 477 sentences and 6030 tokens. Only a few of these tokens can be characterised as PHI.

4 Method

The development of the gold standard involved two main steps: extension and annotation. The gold standard was evaluated by measuring the Inter-Annotator Agreement (IAA) and by testing it on a hybrid de-identification tool.

4.1 Extension

The original dataset, *NorSynthClinical*, contains very few PHIs. Therefore, it was extended with

³NorSynthClinical, <https://github.com/ltagoslo/NorSynthClinical>.

synthetic PHIs (see example below). Where applicable, substatements and single words, or tokens, were manually added to the corpus. Most of the tokens were randomly selected from publicly available lists, such as Statistics Norway’s lists of personal names used by 200 Norwegians or more⁴. The rest of the tokens were invented. They did, however, follow specific Norwegian formats, such as for social security numbers⁵ and phone numbers⁶. For more details regarding the extension, see (Bråten, 2020).

1. Original sentence in Norwegian: *Moren har visstnok noen hjerteproblemer, hun er 75 år gammel.* (The mother apparently has some heart problems, she is 75 years old.)
2. Extended sentence: *Moren har visstnok noen hjerteproblemer, hun er 75 år gammel og bor på Bakklandet Menighets Omsorgsenter.* (The mother apparently has some heart problems, she is 75 years old and lives at Bakklandet Menighets Omsorgsenter.)

4.2 Annotation

The second step of the gold standard development involved annotation. Named Entity Tagging, using the tags provided in Table 1, as proposed by (Dalianis and Velupillai, 2010), was applied in order to mark up elements of PHI. Annotation guidelines were developed⁷, and the tags were assigned in the following way in the following Norwegian sentence:

3. *Moren har visstnok noen hjerteproblemer, hun er <Age>75 år</Age> gammel og bor på <Health_Care_Unit>Bakklandet Menighets Omsorgsenter</Health_Care_Unit>.* In Eng. (The mother apparently has some heart problems, she is <Age>75 years

⁴Norwegian personal names, <https://www.ssb.no/statbank/table/12891/> and <https://www.ssb.no/statbank/table/10501/>

⁵Social security numbers, <https://www.skatteetaten.no/en/person/National-Registry/Birth-and-name-selection/Children-born-in-Norway/National-ID-number/>

⁶Phone numbers, <https://www.nkom.no/telefoni-og-telefonnummer/telefonnummer-og-den-norske-nummerplan/alle-nummerserier-for-norske-telefonnumre>

⁷Annotation guidelines, https://github.com/synnobra/NorSynthClinical-PHI/raw/master/Annotation_guidelines.pdf

```
</Age>old and lives at<Health_Care_Unit>
Bakklandet Menighets Omsorgsenter
</Health_Care_Unit>.)
```

PHI tags

First_Name
Last_Name
Age
Health_Care_Unit
Phone_Number
Social_Security_Number
Date_Full
Date_Part
Location

Table 1: The Named Entity Tag set used to mark up elements of PHI.

Two annotators annotated the whole corpus separately in order to facilitate error detection and comparative evaluation. The annotators, one master of medical science student, A1, and one finance manager, A2, were both Norwegian native speakers. No specific medical knowledge was needed to carry out the annotation.

4.3 Evaluation using Inter-Annotator Agreement and a hybrid de-identification tool

As mentioned, the gold standard was evaluated by measuring the IAA. This is a common evaluation method for providing a quantitative score of how accurate an annotation task is (Pustejovsky and Stubbs, 2012). The two annotated corpora written in UTF-8 encoding format, were converted to CoNLL⁸ format, using a Python3 script, to enable the measurement of IAA. During this process, a token was defined as a string of characters between two spaces or a delimiter. The symbols that were defined as a part of a token, were percentage symbols located to the right of a number as well as hyphens and full stops between two letters or numbers. Moreover, the named entity tags were assigned *IOBES* schema, indicating whether a token was *Inside*, *Outside*, in the *Beginning* or in the *End* of an entity, or whether the entity was represented by a *Single* token, (Collobert et al., 2011). The evaluation metrics used to measure the IAA were precision, recall and F₁-measure.

Further evaluation was conducted by executing the de-identification tool developed for Nor-

⁸CoNLL, Conference on Natural Language Learning

NorNE Label	PHI Tags Label
B-PER	First_Name
I-PER	Last_Name
B-ORG	S/B Health_Care_Unit
I-ORG	I/E Health_Care_Unit
B-LOC	S/B Location
I-LOC	I/E Location

Table 2: NorNE labels matched to PHI Tags labels. S = Single, B = Beginning, E = Ending, I = Inside, O = Outside

wegian pathology reports, employing the same metrics of precision, recall and F_1 -measure as for the IAA. The de-identification tool is a hybrid de-identification tool utilizing a Conditional Random Fields (CRF)⁹ machine learning (ML) model trained on the Bokmål half of the *NorNE* corpus and regular expressions (REGEX) rule-based pattern matching. NorNe is a corpus of Norwegian non-clinical text made publicly available (Jørgensen et al., 2020), The hybrid de-identification tool is further described in (Wie, 2020).

Some, but not all PHI entities in the developed gold standard are found in the *NorNE* training data set. Furthermore, the labels in the *NorNE* data set differ from the gold standard’s PHI both in label names and annotation schema¹⁰. The labels are matched as seen in Table 2¹¹. As the CRF machine learning model is unable to recognize entities not found in the training set, some entities are detected by ML and some by REGEX, see Table 3.

Label	Method
First_Name	CRF
Last_Name	CRF
S/B Health_Care_Unit	CRF
I/E Health_Care_Unit	CRF
Location	CRF
Age	REGEX
Date	REGEX
Phone_Number	REGEX
Social_Security_Number	REGEX

Table 3: Method for detecting labels.

The evaluation done by the de-identification ap-

⁹sklearn-crfsuite, <https://github.com/TeamHG-Memex/sklearn-crfsuite>

¹⁰NorNE uses the IOB2 schema (Jørgensen et al., 2020)

¹¹Most notable is the matching of ORG and Health_Care_Unit

plication is based on the CoNLL format described earlier in this chapter. The de-identification application was not designed to distinguish between Date_Part and Date_Full, so these entities were combined for the evaluation. Furthermore, the REGEX for phone numbers, dates and social security numbers were not designed to recognize entities split into more than one token.

5 Results

5.1 Extension and Annotation

An extended and annotated version of the *NorSynthClinical* corpus has been created. It has been given the name *NorSynthClinical PHI* and made publicly available on GitHub¹². In total, it consists of 8,270 tokens and 409 PHI instances. The distribution of the PHI categories and an overview of the number of tokens added during the extension, is provided in Table 4. Moreover, Figure 1 shows the number and distribution of annotations where the annotators agreed and not, resulting in a micro-averaged overall IAA of 0.94, see Table 5. Only annotations with exactly the same tag and span were considered matching.

5.2 De-identification

The initial evaluation test yielded the results seen in Table 6 - a micro-averaged F_1 -measure of 0.553. Following the initial test, the two following modifications were implemented:

1. The Health_Care_Unit entity label and Location entity label were merged.
2. The labels for entities predicted by rule-based methods were reduced – leaving the single-token instances and the first token in multi-token instances as is, and removing the rest.

These modifications yielded a micro-averaged F-measure of 0.730 and a recall of 0.619, see Table 7, and are discussed further in the analysis and discussion chapter.

6 Analysis

6.1 The NorSynthClinical PHI corpus

The amount of PHI in the extended corpus, *NorSynthClinical PHI*, constitutes around 5% of

¹²NorSynthClinical PHI, <https://github.com/synnobra/NorSynthClinical-PHI>.

PHI category	PHI in the <i>NorSynthClinical</i> corpus	Added PHI	PHI in the gold standard corpus
First_Name	0	70	70
Last_Name	0	49	49
Age	162	0	162
Health_Care_Unit	12	30	42
Phone_Number	0	9	9
Social_Security_Number	0	5	5
Date_Full	0	18	18
Date_Part	46	-1*	45
Location	3	6	9
Total	223	186	409

*A *Date_Part* became a *Date_Full* because additional information was added to the original *Date_Part*.

Table 4: The distribution of PHI categories in the original *NorSynthClinical* corpus containing 7,863 tokens) and the extended *NorSynthClinical PHI* corpus containing 8,270 tokens).

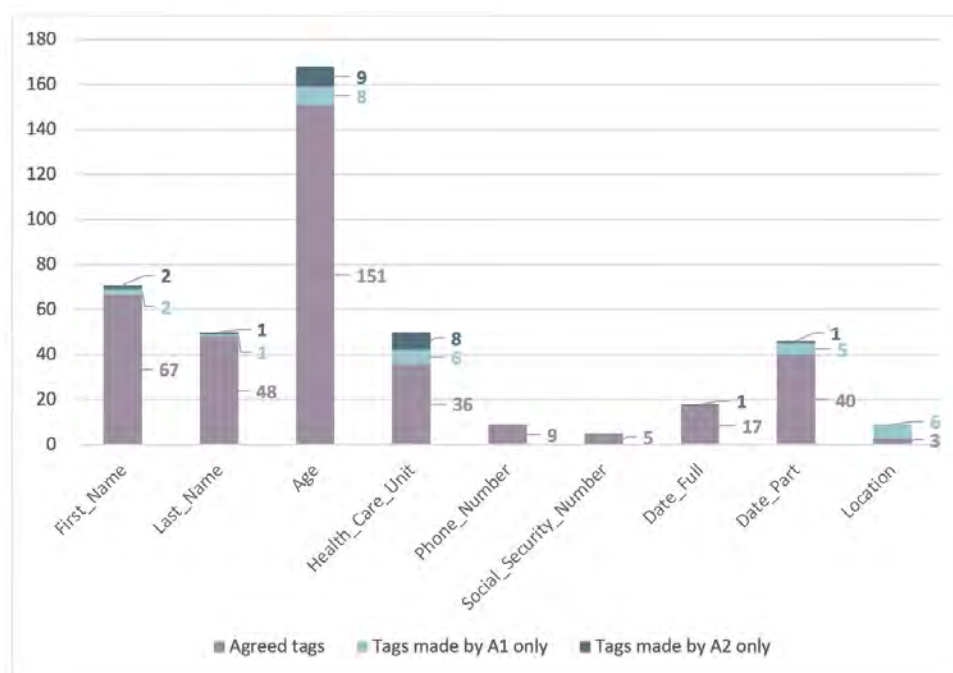


Figure 1: The distribution of agreed (n=376) and disagreed (n=50) annotation tags in each PHI category made by the two annotators A1 and A2

the content. This is above the average of 2% (Dalianis, 2018), but quite similar to the 4.3% reported by Bjurström and Singh (2013). Even the distribution of the different PHI categories resembles the distribution in other clinical texts where names and dates make up the largest categories (Neamatullah et al., 2008; Dalianis and Velupillai, 2010; Deleger et al., 2014; Hanauer et al., 2013). In the extended corpus, names (including

First_Name and Last_Name combined) make up almost one third of the overall PHI, and dates (including Date_Full and Date_Part) more than 15%. The most common category in the extended corpus, however, is Age. In the *NorSynthClinical PHI* corpus, Age constitutes around 39% of all PHI, while in other corpora, it constitutes no more than 1% (Neamatullah et al., 2008; Dalianis and Velupillai, 2010; Deleger et al., 2014).

PHI category	Precision	Recall	F-measure
First_Name	0.97	0.97	0.97
Last_Name	0.98	0.98	0.98
Age	0.94	0.95	0.95
Health_Care_Unit	0.82	0.86	0.84
Phone_Number	1.00	1.00	1.00
Social_Security_Number	1.00	1.00	1.00
Date_Full	0.94	1.00	0.97
Date_Part	0.98	0.89	0.93
Location	1.00	0.33	0.50
Overall performance (micro-averaged)	0.94	0.93	0.94

Table 5: The agreement between the two annotators that annotated the extended *NorSynthClinical* corpus.

Label	Precision	Recall	F ₁ -measure	Support
First_Name	0.951	0.806	0.872	72
Last_Name	0.946	0.964	0.955	55
S/B Health_Care_Unit	0.090	0.167	0.117	42
I/E Health_Care_Unit	0.833	0.192	0.346	26
Location	0.209	1.000	0.346	9
Age (REGEX)	0.985	0.259	0.410	247
Date (REGEX)	0.862	0.770	0.797	74
Social_Security_Number (REGEX)	1.000	0.286	0.444	7
Phone_Number (REGEX)	1.000	0.217	0.357	23
Micro avg.	0.675	0.468	0.553	555

Table 6: Initial evaluation test with the hybrid de-identification tool.

Label	Precision	Recall	F ₁ -measure	Support
First_Name	0.951	0.806	0.872	72
Last_Name	0.946	0.964	0.955	55
S/B Health_Care_Unit	0.767	0.647	0.702	51
I/E Health_Care_Unit	1.000	0.231	0.375	26
Age (REGEX)	0.985	0.395	0.564	162
Date (REGEX)	0.783	0.857	0.818	63
Social_Security_Number (REGEX)	1.000	0.400	0.571	5
Phone_Number (REGEX)	0.800	0.444	0.571	9
Micro avg.	0.893	0.619	0.731	443

Table 7: Final evaluation with the modified hybrid de-identification tool. The entities Health_Care_Unit and Location observed in Table 6 were merged into Health_Care_Unit. in this table

6.2 Inter-Annotator Agreement

The IAA score of 0.94 indicates that the agreement between the two annotators is high. This is especially true for the categories Phone_Number and Social_Security_Number, which the annotators completely agreed on, see Figure 1. How-

ever, these and most other categories contain a small number of PHI instances, questioning the reliability of the statistical analysis. The categories that the annotators disagreed on the most, were Health_Care_Unit and Location, see Figure 1. On five occasions, a PHI instance was anno-

tated as Location by one of the annotators and as Health_Care_Unit by the other annotator. Other disagreements were caused by differences in the annotation span or in the interpretations of the provided annotation guidelines.

6.3 Evaluation with a hybrid de-identification tool

While the overall score for First_Name and Last_Name was high, the scores for Health_Care_Unit and Location were low, see Table 6. The low scores were suspected to be due to health care units often being named after locations and being syntactically similar, resulting in the CRF model frequently labelling Health_Care_Unit as Location – which was confirmed with a manual review of the incorrect predictions.

“Of the 35 incorrect predictions where the correct label was B-ORG, 24 were labelled as B-LOC (approx. 69%).

Of the 21 incorrect predictions where the correct label was I-ORG, 6 were labelled as I-LOC (approx. 29%).” (Wie, 2020)

For the entities processed by the rule-based part (REGEX) of the hybrid de-identification tool the initial precision was high (0.908 micro avg.). However, the recall was low for all entities except date (0.770). This was attributed to the CoNLL conversion of the *NorSynthClinical PHI* corpus splitting the pertinent entities into more tokens, which the de-identification application was not designed to handle. Another consequence of some of these entities being split is an inflation of the support for these categories. An example being the original nine instances of Phone_Number in the *NorSynthClinical PHI* corpus being counted as 23 instances – skewing the recall score, see Table 8. Applying the aforementioned modification of reduction based on prefixes resulted in the same instance support as the original.

7 Discussion and conclusion

What makes the *NorSynthClinical PHI* special and valuable is the fact that it is synthetic. As it does not contain any real personal information, the gold standard can be accessed by anyone and utilized in the development of tools for de-identification of Norwegian clinical text. Hope-

Label	Original	CoNLL
Age	162	247
Date	63	74
Social_Security_Number	5	7
Phone_Number	9	23

Table 8: Converting from SGML format to CoNLL format support inflation.

fully, this will facilitate more research on the content of clinical notes, and eventually a better health care.

The major weakness of the created gold standard is its small size. The English corpus *MIMIC II* consists of 412,509 clinical notes and the *Stockholm EPR PHI* corpus consists of 100 patient records (Dalianis, 2018). As mentioned in (Velupillai et al., 2009), the latter contributes 174,000 tokens. In comparison, the *NorSynthClinical PHI*, which consists of 8,270 tokens, is very small. Besides, it is very specific to the area of cardiology, written by one cardiologist, and extended by a layman. Therefore, there might be a lack of linguistic variety. Furthermore, the gold standard is written in Norwegian Bokmål and not in Nynorsk. However, it would be relatively uncomplicated to translate the gold standard from Bokmål to Nynorsk.

The de-identification tool used for evaluating *NorSynthClinical PHI* corpus was initially designed for another purpose¹³ and trained on publicly available data. The effect of fundamental incompatibilities between the training set and the gold standard, like the disparity between *ORG* and *Health_Care_Unit*, is difficult to estimate. However, no other de-identification system for Norwegian is available.

The final evaluation of the modified hybrid de-identification tool for Norwegian using *NorSynthClinical PHI* gave an F_1 -measure of 0.731 and a recall of 0.619.

A de-identification tool is aiming on a higher recall to remove all possible PHIs, also on the cost of lower precision.

Further improvements could be made to the de-identification tool. Implementing dictionary-based algorithms could improve the accuracy of certain entity types. Task-specific dictionaries for Norwegian health care units and/or medications are feasible implementations and would

¹³De-identification of Norwegian pathology reports.

likely improve accuracy on clinical texts. Furthermore, implementing tokenization directly in the de-identification tool would allow for de-identification of untokenized text, and minimize incompatibilities between the input and de-identification algorithm.

The gold standard has its limitations and cannot alone decide whether a specific tool provides sufficiently de-identified outcomes. Therefore, we encourage to further expansions of the gold standard corpus, in addition to more evaluation research, in order to make it more reliable and improve its quality.

Contributions of each author

SB made and evaluated the gold standard corpus, and wrote in the article. WW developed the hybrid de-identification tool and tested it on the gold standard corpus and co-authored the paper. HD supervised the study, gave comments and wrote in the article.

References

- Roar Bjurstrøm and Jaspreet Singh. 2013. De-identification of Norwegian Health Record Notes: An Experimental Approach. Master's thesis, Institutt for datateknikk og informasjonsvitenskap.
- Synnøve Bråten. 2020. Extending a Synthetic Norwegian Clinical Corpus for De-Identification. Master's thesis, Department of Computer and Systems Sciences, Stockholm University and Karolinska Institutet, <https://daisy.dsv.su.se/fil/visa?id=230054>.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12:2493–2537.
- Hercules Dalianis. 2018. *Clinical text mining: Secondary use of electronic patient records*. Springer Nature, Open Access.
- Hercules Dalianis. 2019. Pseudonymisation of Swedish Electronic Patient Records Using a Rule-Based Approach. In *Proceedings of the Workshop on NLP and Pseudonymisation*, pages 16–23, Turku, Finland. Linköping Electronic Press.
- Hercules Dalianis and Sumithra Velupillai. 2010. De-identifying Swedish clinical text-refinement of a gold standard and experiments with Conditional random fields. *Journal of Biomedical Semantics*, 1(1):6.
- Louise Deleger, Todd Lingren, Yizhao Ni, Megan Kaiser, Laura Stoutenborough, Keith Marsolo, Michal Kouril, Katalin Molnar, and Imre Solti. 2014. Preparing an annotated gold standard corpus to share with extramural investigators for de-identification research. *Journal of Biomedical Informatics*, 50:173–183.
- Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. 2017. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606.
- Khaled El Emam, Fida Kamal Dankar, Romeo Issa, Elizabeth Jonker, Daniel Amyot, Elise Cogo, Jean-Pierre Corriveau, Mark Walker, Sadrul Chowdhury, Regis Vaillancourt, et al. 2009. A globally optimal k-anonymity method for the de-identification of health data. *Journal of the American Medical Informatics Association*, 16(5):670–682.
- David Hanauer, John Aberdeen, Samuel Bayer, Benjamin Wellner, Cheryl Clark, Kai Zheng, and Lynette Hirschman. 2013. Bootstrapping a de-identification system for narrative patient records: cost-performance tradeoffs. *International Journal of Medical Informatics*, 82(9):821–831.
- Fredrik Jørgensen, Tobias Aasmoe, Anne-Stine Ruud Husevåg, Lilja Øvrelid, and Erik Velldal. 2020. <https://www.aclweb.org/anthology/2020.lrec-1.559> NorNE: Annotating named entities for Norwegian. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4547–4556, Marseille, France. European Language Resources Association.
- Montserrat Marimon, Aitor Gonzalez-Agirre, Ander Intxaurreondo, Heidy Rodriguez, Jose Lopez Martin, Marta Villegas, and Martin Krallinger. 2019. Automatic De-identification of Medical Texts in Spanish: the MEDDOCAN Track, Corpus, Guidelines, Methods and Evaluation of Results. In *IberLEF@ SE-PLN, La Sociedad Española para el Procesamiento del Lenguaje Natural*, pages 618–638.
- Ishna Neamatullah, Margaret M Douglass, H Lehman Li-wei, Andrew Reisner, Mauricio Villarroel, William J Long, Peter Szolovits, George B Moody, Roger G Mark, and Gari D Clifford. 2008. Automated de-identification of free-text medical records. *BMC Medical Informatics and Decision Making*, 8(1):32.
- James Pustejovsky and Amber Stubbs. 2012. *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. O'Reilly Media, Inc.
- Taraka Rama, Pål Brekke, Øystein Nytrø, and Lilja Øvrelid. 2018. Iterative development of family history annotation guidelines using a synthetic corpus of clinical text. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 111–121.

Georg Rehm and Hans Uszkoreit. 2012. The Norwegian Language in the European Information Society. In *The Norwegian Language in the Digital Age*, pages 45–51. Springer.

Mohammed Saeed, Christine Lieu, Greg Raber, and Roger G Mark. 2002. MIMIC II: a massive temporal ICU patient database to support research in intelligent patient monitoring. In *Computers in cardiology*, pages 641–644. IEEE.

Sumithra Velupillai, Hercules Dalianis, Martin Hassel, and Gunnar H Nilsson. 2009. Developing a standard for de-identifying electronic patient records written in Swedish: precision, recall and F-measure in a manual and computerized annotation trial. *International Journal of Medical Informatics*, 78(12):e19–e26.

Wilhelm Wie. 2020. De-identification of Norwegian Clinical Text A Hybrid Approach Using Publicly Available Data. Master’s thesis, Department of Computer and Systems Sciences, Stockholm University, <https://daisy.dsv.su.se/fil/visa?id=230198>.

Applying and Sharing pre-trained BERT-models for Named Entity Recognition and Classification in Swedish Electronic Patient Records

Mila Grancharova

Department of Computer
and Systems Sciences
Stockholm University
Kista, Sweden

`mila.grant@gmail.com`

Hercules Dalianis

Department of Computer
and Systems Sciences
Stockholm University
Kista, Sweden

`hercules@dsv.su.se`

Abstract

To be able to share the valuable information in electronic patient records (EPR) they first need to be de-identified in order to protect the privacy of their subjects. Named entity recognition and classification (NERC) is an important part of this process. In recent years, general-purpose language models pre-trained on large amounts of data, in particular BERT, have achieved state of the art results in NERC, among other NLP tasks. So far, however, no attempts have been made at applying BERT for NERC on Swedish EPR data.

This study attempts to fine-tune one Swedish BERT-model and one multilingual BERT-model for NERC on a Swedish EPR corpus. The aim is to assess the applicability of BERT-models for this task as well as to compare the two models in a domain-specific Swedish language task. With the Swedish model, recall of 0.9220 and precision of 0.9226 is achieved. This is an improvement to previous results on the same corpus since the high recall does not sacrifice precision. As the models also perform relatively well when fine-tuned with pseudonymised data, it is concluded that there is good potential in using this method in a shareable de-identification system for Swedish clinical text.

1 Introduction

Electronic patient records (EPR), also called clinical text, contain valuable information about patients' symptoms, physicians' assessments, diagnoses, treatments and treatment outcomes. Advancements in natural language processing (NLP) and machine learning have made it possible to use large amounts of clinical text to assist physicians

and medical researchers in detecting early symptoms of disorders, predicting adverse effects of treatments, etc, see Chapter 10 in (Dalianis, 2018). However, clinical text contains information that can reveal the identity of patients and other mentioned individuals, so called Protected Health Information (PHI). Methods have been developed to detect this information and obscure it in order to protect people's identities (Meystre et al., 2010; Stubbs et al., 2015). One important note to make is that de-identified text cannot be guaranteed to be safe to release and must still be handled with great care. A good de-identification system can, however, help facilitate an efficient anonymisation process.

In this study PHI refers only to the named entities which may reveal a person's identity, such as name, age and location. In this sense, detecting and identifying the PHI before obscuring it is a Named Entity Recognition and Classification (NERC) problem. When it comes to data-driven NERC, models based on recurrent neural networks (RNNs) and long short-term memory (LSTM) networks have been successfully used for several languages (Lê et al., 2020; Lange et al., 2019). In the last two years, however, transformer-based language models such as BERT have achieved state-of-the-art results in several NLP task on commonly used data sets (Devlin et al., 2019).

BERT is a general-purpose language model developed by Devlin et al. (2019). In essence, BERT is a neural network based on transformers. Transformers are a type of deep learning model designed to handle sequential data, such as natural language text. Since their introduction in 2017 (Vaswani et al., 2017), transformers have been widely used across a variety of NLP tasks, not least on clinical text (Lewis et al., 2020). The benefit of transformer-based models over previous architectures is that they do not require the sequential data to be processed in order, allowing for parallelization of the training process. This has made it possible to

develop large pre-trained models such as BERT, which have been fitted on larger amounts of data than was previously feasible.

Since the first BERT-model was released in 2018, several models with modified architecture and different data used in pre-training have been released, including the multilingual M-BERT¹. M-BERT is pre-trained on texts in 104 languages, including Swedish. In 2019, the National Library of Sweden released a Swedish BERT model, KB-BERT², pre-trained exclusively on Swedish texts.

To use a pre-trained BERT-model for a downstream task, it needs to be fine-tuned for that task. Both KB-BERT and M-BERT have shown success in the NERC task for Swedish when fine-tuned with the publicly available Stockholm-Umeå Corpus consisting of Swedish texts from the 1990's (Malmsten et al., 2020). To our knowledge, however, no previous attempt has been made at using these models for NERC in Swedish EPR data.

In this study, we attempt to improve NERC performance on Swedish electronic patient records by fine-tuning KB-BERT and M-BERT with domain-specific data. More specifically, our aim is to achieve high recall, which is a priority in the de-identification task, without sacrificing precision. A risk with de-identification methods based on machine learning is that a model trained on sensitive data could be re-engineered, revealing the data. In a BERT-model, there are no links between words in the vocabulary, making it infeasible to retrieve the patient records used for fine-tuning. However, due to names and other personal identifiers appearing in the model's vocabulary, there may be legal issues with releasing a model fine-tuned on patient records. Therefore, in an additional experiment, the models are fine-tuned using pseudonymised patient records to see how NERC performance on authentic records is affected.

The outline of this paper is as follows. First, Section 2 presents some previous studies on NERC in clinical text and specifically previous results on the data set at hand. Then, Section 3 describes the data used in this study, gives some more detail on the two BERT models and goes through how the fine-tuning and evaluation are performed. Section 4 presents the results for both models. Finally, the results are discussed in Section 5.

¹M-BERT, <https://github.com/google-research/bert>

²KB-BERT, <https://github.com/Kungbib/swedish-bert-models>

2 Related Research

There are several publicly available BERT-models pre-trained specifically for the biomedical and clinical domains. In 2019, Lewis et al. (2020) released BioBERT³, a BERT-model pre-trained on PubMed articles as well as Wikipedia articles and books. The authors present an F₁-score of approximately 0.87 on the commonly used i2b2 2010 data set for clinical text NERC. In a different 2019 project, (Peng et al., 2019) continued to pre-train the pre-trained BERT-model released by (Devlin et al., 2019) on PubMed abstracts and clinical notes. This model, named BlueBERT⁴, reaches an F₁-score of approximately 0.77 on the i2b2 data set. The same year, (Alsentzer et al., 2019) released clinicalBERT⁵ pre-trained on clinical texts but also specifically on discharge summaries. The combined Bio+Discharge Summary model reaches an F₁-score of 0.88 on the i2b2 2010 data set. All of these models are only pre-trained on English texts.

For non-English clinical text NERC, some advancements were made in connection to the 2019 shared task MEDDOCAN which consisted of performing NERC on Spanish electronic patient records with annotated PHI. In a submission to the contest Mao and Liu (2019) used M-BERT, which is also pre-trained on Spanish text (Mao and Liu, 2019) with a decoding CRF layer for token classification. They also applied some post-processing techniques, achieving F₁-score and recall of approximately 0.93.

When it comes to Swedish, several attempts have been made at performing NERC on the annotated data set of electronic patient records Stockholm EPR PHI Corpus. In one study by Berg and Dalianis (2020) the authors extended the annotated data set with data generated using a semi-supervised learning method with the aim of increasing recall without sacrificing precision. The highest recall reported was 0.8920, at which point the precision was 0.9420. These results were achieved using a Conditional Random Field (CRF) model. Grancharova et al. (2020) managed to increase the recall to 0.9209 using the same model by under-sampling negative tokens, thus tokens not belonging to a PHI. However, this came at the cost of significant

³BioBERT, <https://github.com/dmis-lab/biobert>

⁴BlueBERT, <https://github.com/ncbi-nlp/bluebert>

⁵clinicalBERT, <https://github.com/EmilyAlsentzer/clinicalBERT>

decrease in precision to 0.8819. Regarding the application of models trained on pseudonymised clinical data for NERC on authentic data, there is a study by Berg et al. (2019) where the authors achieved at highest recall of 0.5510 using a LSTM network. The experiment was repeated with a classic CRF and the recall decreased to 0.4983.

3 Data and Methods

This section describes the data, tools and methods used in this study. First, the EPR data set is described in Section 3.1. Then, Section 3.2 describes the BERT-models used and how they were fine-tuned. Lastly, Section 3.3 describes how the models were evaluated in a number of experiments.

3.1 Data

The data used in this study is Stockholm EPR PHI Corpus⁶ Stockholm EPR PHI Corpus is part of the research infrastructure Health Bank - The Swedish Health Records Research Bank⁷. Stockholm EPR PHI Corpus consists of 200,000 tokens with nine annotated PHI classes. See Table 1 for the classes and their distribution.

The annotation of Stockholm EPR PHI Corpus is described in more detail in (Velupillai et al., 2009). The data was refined in the first de-identification experiment described in (Dalianis and Velupillai, 2010) and has since been used in several studies. Figure 1 shows an example of an pseudonymised annotated record from the data set, followed by an English translation of the same record.

When formatting the data for fine-tuning, tagged entities consisting of multiple words were split into separate tokens and tagged according to the BIOES-standard. This means marking whether a positive token is in the beginning ('B'), ending ('E') or inside ('I') a named entity, or if the token itself makes up a named entity ('S') (Reimers and Gurevych, 2017). Negative tokens, thus tokens which are not part of a named entity, were marked 'O'.

3.2 Methods

This section describes the methods used in this study. First, Section 3.2.1 gives more details on the two pre-trained BERT models used. Then, Section 3.2.2 describes how the models were fine-tuned.

⁶This research has been approved by the Swedish Ethical Review Authority under permission no. 2019-05679.

⁷Health Bank, <http://dsv.su.se/healthbank>

PHI Class	Instances
First Name	923
Last Name	931
Phone Number	137
Age	55
Full Date	457
Date Part	709
Health Care Unit	1,414
Location	95
Organisation	43
Total	4,764

Table 1: The class distribution of Stockholm EPR PHI Corpus.

3.2.1 BERT models

The BERT-models used in this study are the Swedish KB-BERT and the multilingual M-BERT. Both models implement the BERT-Base architecture consisting of twelve layers with a hidden size of 768 and $11 \cdot 10^7$ parameters.

KB-BERT was released by the National Library of Sweden in 2019 (Malmsten et al., 2020). It is pre-trained on approximately 20 GB of digitized Swedish texts written between the years 1940 and 2019. The resources include news articles, legal text, social media posts and Swedish Wikipedia articles. This results in a vocabulary size of around 50,000 tokens. The model is cased, meaning that there are separate entries for tokens beginning with an upper case letter and tokens beginning with a lower case letter.

Devlin et al. (2019) released a multilingual BERT model alongside the original English BERT model. The multilingual model used in this study, M-BERT, is the cased version of this model. It has been pre-trained on 104 languages, including Swedish. For each language, the training data consisted of Wikipedia articles written in that language. To balance the data, high-resource languages were under-sampled while low-resource languages were over-sampled using exponentially smoothed weighting of the data. M-BERT has a vocabulary size of around 120,000 tokens.

3.2.2 Fine-tuning

The pre-trained BERT models provide a general representation, or encoding, of input data. To use the models for prediction or inference they need to be fine-tuned for a specific down-stream task. This involves adding an additional output layer and fit-

Planeringsansvarig: SSK Tjänstgörande
Patientansvarig läkare: <First_Name>Mohamed</First_Name>
<Last_Name>Åström</Last_Name>
Kontaktorsak: Ramlat i hemmet <Full_Date>10/5-2006</Full_Date> och krampat
<Date_Part>12/5</Date_Part>.
Hade inte ätit eller druckit på 4 dygn.
Hälsohistoria: vårderf. Se läkare anteckningar.
Närstående: Dotter <First_Name>Jessica</First_Name><Last_Name>Fredriksson</Last_
Name> tel: <Phone_Number>0715-463920</Phone_Number>,
tel hem <Phone_Number>92 35 45</Phone_Number> <Last_Name>Fredriksson</Last_Name>
tel. <Phone_Number>0392-857461</Phone_Number>
Social bakgrund: Bor på gruppboende, <Health_Care_Unit>Lärkan</Health_Care_Unit>
på <Location>Ladugårdsgärdet</Location>.

Planning manager: Nurse on duty
Attending physician: <First_Name>Mohamed</First_Name>
<Last_Name>Åström</Last_Name>
Reason of contact: Fallen at home <Full_Date>10/5-2006</Full_Date> and felt cramps
<Date_Part>12/5</Date_Part>.
Had not eaten or drunk in 4 days.
Health background: See physician's notes.
Family: Daughter <First_Name>Jessica</First_Name><Last_Name>Fredriksson</Last_
Name> ph: <Phone_Number>0715-463920</Phone_Number>,
ph. home <Phone_Number>92 35 45</Phone_Number>
<Last_Name>Fredriksson</Last_Name>
ph. <Phone_Number>0392-857461</Phone_Number>
Social background: Lives at nursing home,
<Health_Care_Unit>Lärkan</Health_Care_Unit> at <Location>Ladugårdsgärdet</Location>.

Figure 1: Example of a pseudonymised electronic patient record in Swedish from Stockholm EPR PHI Corpus and its translation to English.

ting the model with task-specific data. In this case, the down-stream task is NERC and the data used for fine-tuning is that described in Section 3.1. The pre-trained models were loaded and fine-tuned using the *HuggingFace's Transformers library* (Wolf et al., 2020). Both models were loaded with the library's *BertForTokenClassification structure* which provides a linear output layer on top of the hidden-states output.

A challenge with fine-tuning BERT is hyper-parameter optimization. The model is sensitive to several parameters such as number of epochs, batch size and learning rate. Devlin et al. (2019) found that for large data sets the hyper-parameters do not have great impact on performance. On smaller data sets, the authors recommend performing some hyper-parameter optimization for the task at hand. Due to the size of the models, the time it takes

to fine-tune them presents a limit on how much resources can be delegated to hyper-parameter optimization. In this study, the optimization is limited to a simple parameter search with starting point at the values recommended by Devlin et al. (2019).

3.3 Application of methods: Experiments

This section presents the different experiments performed to generate the results presented in this paper. First, 20% of the original data, selected at random, was held out for testing. Out of the remaining data, 20% was reserved for development. The purpose of the development set was to evaluate different hyper-parameter settings. The remaining data, which we call the training set, was used for fine-tuning.

In addition to the original training set, Stockholm EPR PHI Corpus, we created a version of the

training set, Stockholm EPR PHI Pseudo Corpus, where the PHIs have been replaced by surrogates. We call this the pseudonymised training set⁸, or *pseudo* for short.

The surrogate generation is lexical, based on the collection of Swedish named entity lists used in (Dalianis, 2019). In this study, however, the variation of surrogate names is much larger, containing 123,000 female first names, 121,000 male first names and 35,000 last names, rather than only the 100 most common first- and last names used in (Dalianis, 2019).

After fine-tuning on the pseudonymised training set, the models were evaluated on the original test set. The motivation behind these tests is that models trained on pseudonymised data are safer to release for further development by other researchers, without risking that the PHI is revealed. Therefore, it is of interest to see how well such models perform on authentic, not de-identified, patient records.

For both KB-BERT and M-BERT, a search over hyper-parameters was performed. The batch size was set to 16 and the learning rate to $5 \cdot 10^{-5}$. When it comes to the number of epochs, the results differed slightly between the models. Figures 2 and 3 show the precision and recall for different number of epochs over the training set when fine-tuning KB-BERT and M-BERT, respectively. When choosing the number of epochs, most attention was paid to recall as that is of highest priority in a de-identification system. For all models except one, recall either decreased or did not improve significantly after three epochs. Thus, the models were fine-tuned for three epochs. The exception was M-BERT fitted with the pseudonymised data which was fine-tuned for four epochs. The precision was also monitored and it was observed, as the figures show, that precision continued to increase longer than recall. Since recall was prioritized and resources were limited, no experiments were made with training the models further.

After the models were fine-tuned, they were evaluated on the original test set, namely the held out data set, 20% of Stockholm EPR PHI Corpus. We call this set *test set A*. In order to test how well the models perform on a broader range of EPR data, they were also evaluated on other medical specialties of Swedish EPR Corpora from Health Bank. For the purpose of this report, this second test set is

⁸Generally, most research on clinical text is carried out on pseudonymised data while most studies on Health Bank data have used real data.

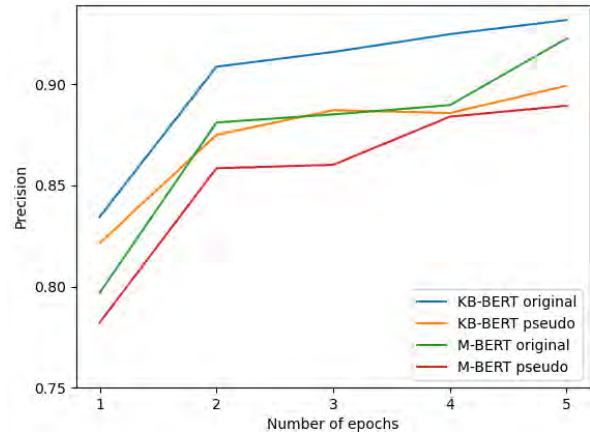


Figure 2: Precision on the development set after different number of epochs for all four models.

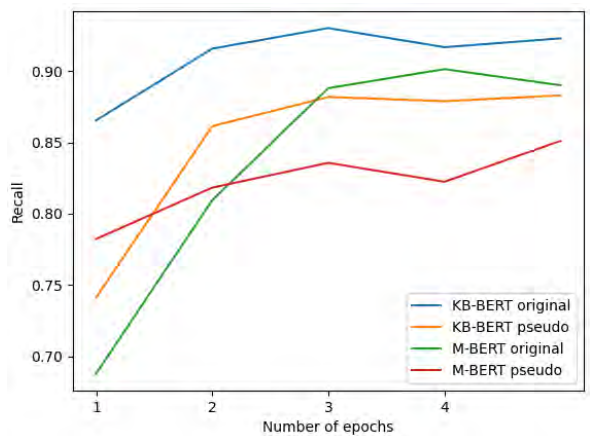


Figure 3: Recall on the development set after different number of epochs for all four models.

called *test set B*. For the most part, *test set B* is annotated according to the same standard as Stockholm EPR PHI Corpus but is lacking the *Organisation* class which is thus excluded from the evaluation on this test set. Further, *test set B* contains ages and dates but their annotation differs from those in Stockholm EPR PHI Corpus. In order to minimize the error caused by different annotation standards, the classes *Age*, *Date Part* and *Full Date* are also excluded from evaluation.

4 Results

The results presented in this section were achieved with the best hyper-parameter values found, see Section 3.3. Note that the hyper-parameter optimization was not exhaustive and this may have significant effects on the results.

Table 2 shows the precision (P), recall (R) and F₁-score for the two models fine-tuned with the

original training set, as well as those fine-tuned on the pseudonymised training set, when evaluated on *test set A*. Table 3 shows the corresponding scores for *test set B*.

Model	Data	P	R	F ₁
KB	Original	0.9226	0.9220	0.9223
	Pseudo	0.8827	0.8822	0.8824
M	Original	0.9051	0.8899	0.8974
	Pseudo	0.8602	0.8357	0.8478

Table 2: Precision, recall and F₁-score of KB-BERT and M-BERT fine-tuned with the original training set and the pseudonymised training set respectively, and evaluated on *test set A*.

Model	Data	P	R	F ₁
KB	Original	0.6923	0.7272	0.7093
	Pseudo	0.6427	0.7439	0.6896
M	Original	0.6494	0.6847	0.6666
	Pseudo	0.6398	0.6963	0.6669

Table 3: Precision, recall and F₁-score of KB-BERT and M-BERT fine-tuned with the original training set and the pseudonymised training set respectively, and evaluated on *test set B*.

Table 4 shows the recall per class for the models fine-tuned with the original training set and evaluated on *test set A*. Table 5 shows the corresponding results for *test set B*. In the same manner, Tables 6 and 7 shows the recall per class for all the models fine-tuned with the pseudonymised training set and evaluated on *test set A* and *test set B*, respectively. Note that the averages in all tables are weighted based on the number of instances from each class present in the test set at hand. The number of instances per class are given by the figures within the parentheses in the tables’ first column.

5 Discussion and Conclusions

The results show that the fine-tuned KB-BERT achieves recall on the same level as that reported in (Grancharova et al., 2020) on the same data set, see Table 2. In this study, however, the relatively high recall does not come at the price of low precision. The precision achieved using KB-BERT is on par with the highest recorded precision on Stockholm EPR PHI Corpus which was documented in (Berg and Dalianis, 2020). There, again, recall was below 0.9. Thus, the BERT-model seems to offer a good

Class (instances)	KB-BERT	M-BERT
First Name (195)	0.9385	0.9077
Last Name (213)	0.9531	0.9296
Phone Number (21)	0.9048	0.8571
Age (9)	1.0000	0.7778
Full Date (83)	0.9518	0.9518
Date Part (131)	0.9847	0.9824
Health Care Unit (293)	0.8737	0.8396
Location (19)	0.7895	0.4221
Organisation (10)	0.5000	0.5000
Weighted average	0.9220	0.8899

Table 4: Recall per class of the models fine-tuned with the original training set and evaluated on *test set A*.

Class (instances)	KB-BERT	M-BERT
First Name(208)	0.7212	0.7596
Last Name (282)	0.7270	0.6915
Phone Number (22)	0.8636	0.7727
Health Care Unit (208)	0.7163	0.6394
Location (57)	0.7368	0.5088
Weighted average	0.7272	0.6847

Table 5: Recall per class of the models fine-tuned with the original training set and evaluated on *test set B*.

Class (instances)	KB-BERT	M-BERT
First Name (195)	0.9128	0.8564
Last Name (213)	0.9202	0.8638
Phone Number (21)	0.8095	0.9048
Age (9)	1.0000	0.8889
Full Date (83)	0.9398	0.8554
Date Part (131)	0.9695	0.9847
Health Care Unit (293)	0.8029	0.7577
Location (19)	0.6842	0.4737
Organisation (10)	0.6000	0.5000
Weighted average	0.8822	0.8357

Table 6: Recall per class of the models fine-tuned with the pseudonymised version of the training set and evaluated on *test set A*.

balance between precision and recall. From a pure de-identification perspective, high precision is not a priority. However, for the de-identified data to be of use to physicians and researchers, precision remains important. In this sense, the results presented in this paper can be considered an overall improvement of NERC on this data.

Class (instances)	KB-BERT	M-BERT
First Name(208)	0.8077	0.7548
Last Name (282)	0.7447	0.7092
Phone Number (22)	0.7273	0.7273
Health Care Unit (208)	0.6490	0.6731
Location (57)	0.7368	0.4912
Weighted average	0.7349	0.6963

Table 7: Recall per class of the models fine-tuned with the pseudonymised version of the training set and evaluated on *test set B*.

Regarding the comparison between KB-BERT and M-BERT, the first achieves higher precision and recall on both test sets, see Tables 2 and 3. The difference is more prevalent in some PHI classes than in others. For instance, the recall on *Location* drops significantly when using M-BERT compared to using KB-BERT. This suggests that pre-training specialized toward one language is more beneficial than broader pre-training. This is only a speculation since there are other differences between the two models that could affect performance on the task at hand, such as the nature and amount of Swedish texts used in pre-training.

It is also worth mentioning that the difference in recall between the two models is small, averaging at approximately 0.5 percentage points when fine-tuning on the original data and 1 percentage point when fine-tuning on the pseudonymised data. Since only a limited amount of time was spent on optimisation, it is possible that M-BERT could achieve results similar to KB-BERT if fine-tuned with better settings or more data.

Tables 2 and 3 also show that the models fine-tuned on the original records perform better than those fine-tuned on the pseudonymised records. This is not surprising, as the surrogates have limited range compared to the authentic named entities. Tables 4 - 7 show that, for instance, the recall on *Age* is more negatively affected by fine-tuning on pseudonymised records than the recall on *First name* and *Last name*. An explanation could be that the formats in which surrogate ages are given do not cover all formats present in the authentic records, resulting in greater discrepancies between the training set and the test set when fine-tuning with the pseudonymised records. The formats of names, on the other hand, are less varied in this domain.

Although the models fine-tuned with

pseudonymised data perform worse overall, the differences between them and the same models fine-tuned with the original data are not huge. In some cases, such as *Phone number* in M-BERT, the pseudonymised model actually performs better, see Tables 4 and 6. It is clear that the BERT-models are less sensitive to the discrepancies between the original and pseudonymised data than the CRF and LSTM models used on this data set previously, see Section 2 Related research and (Berg et al., 2019). This suggests that this method should be explored further for the purpose of being able to share models trained on electronic patient records while reducing the risks of breaching the privacy of patients or other individuals mentioned in the text.

A comparison between Table 2 and Table 3 demonstrates that there is a loss in recall and an even greater loss in precision when applying the models to data in a slightly broader domain. Differences in the annotation of the two test sets make a direct comparison difficult, but it is clear that the models have learned enough to generalize relatively well to a broader range of electronic patient records. Future work includes creating more annotated data for evaluation as well as training on a broader range of records in order to improve generalization.

In summary, this paper presents an improvement on previous results on the Stockholm EPR PHI Corpus in the sense that the same high recall is achieved without sacrificing precision. It is also demonstrated that performance is somewhat negatively affected by fine-tuning on pseudonymised electronic patient records but the models still achieve relatively high recall. Due to the benefit of being able to share non-sensitive models in compliance with preserving the privacy of patients, this approach should be studied and developed further. The results also show that KB-BERT outperforms M-BERT overall but both models perform relatively well. We can not make any concrete conclusions on the limitations of the models due to the limited resources delegated to optimisation and the limited data used for fine-tuning. Future work includes optimising the models further and fine-tuning on a larger data set.

On a final note, even with a de-identification system with high recall, the de-identified data could be re-identified using external sources. Therefore, the de-identified data must be handled with care. To improve the privacy where there could be some

false negatives, thus missed PHI, one could remove the tags of the true positive so the false negatives are not distinguishable, performing what is known as HIPS (Hide In Plain Sight) (Carrell et al., 2013).

Acknowledgments

We are grateful to the DataLEASH project for funding this research work. Great thanks also to John Valik Karlsson, M.D., for the assistance with the translation of the patient record to English.

References

- Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, WA Redmond, and Matthew BA McDermott. 2019. Publicly Available Clinical BERT Embeddings. *NAACL HLT 2019*, page 72.
- Hanna Berg, Taridzo Chomutare, and Hercules Dalianis. 2019. Building a De-identification System for Real Swedish Clinical Text Using Pseudonymised Clinical Text. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 118–125.
- Hanna Berg and Hercules Dalianis. 2020. A Semi-supervised Approach for De-identification of Swedish Clinical Text. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille*, pages 4444–4450.
- David Carrell, Bradley Malin, John Aberdeen, Samuel Bayer, Cheryl Clark, Ben Wellner, and Lynette Hirschman. 2013. Hiding in plain sight: use of realistic surrogates to reduce exposure of protected health information in clinical text. *Journal of the American Medical Informatics Association*, 20(2):342–348.
- Hercules Dalianis. 2018. *Clinical text mining: Secondary use of electronic patient records*. Springer.
- Hercules Dalianis. 2019. Pseudonymisation of Swedish Electronic Patient Records Using a Rule-Based Approach. In *Proceedings of the Workshop on NLP and Pseudonymisation*, pages 16–23, Turku, Finland. Linköping Electronic Press.
- Hercules Dalianis and Sumithra Velupillai. 2010. De-identifying Swedish Clinical Text - Refinement of a Gold Standard and Experiments with Conditional Random Fields. *Journal of Biomedical Semantics*, 1:6.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. <http://arxiv.org/abs/1810.04805> BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *arXiv*, <https://arxiv.org/abs/1810.04805>.
- Mila Grancharova, Hanna Berg, and Hercules Dalianis. 2020. Improving Named Entity Recognition and Classification in Class Imbalanced Swedish Electronic Patient Records through Resampling. In *Proceedings of Eighth Swedish Language Technology Conference (SLTC) 2020, Göteborg, Sweden*.
- Lukas Lange, Heike Adel, and Jannik Strötgen. 2019. NLNDE: The Neither-Language-Nor-Domain-Experts’ Way of Spanish Medical Document De-Identification. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019)*, pages 671–678.
- Patrick Lewis, Myle Ott, Jingfei Du, and Veselin Stoyanov. 2020. Pretrained Language Models for Biomedical and Clinical Tasks: Understanding and Extending the State-of-the-Art. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 146–157.
- Ngoc C. Lê, Ngoc-Ye Nguyen, Anh-Duong Trinh, and Hue Vu. 2020. On the Vietnamese Name Entity Recognition: A Deep Learning Method Approach. In *IEEE Access*.
- Martin Malmsten, Love Börjesson, and Chris Haffenden. 2020. <http://arxiv.org/abs/2007.01658> Playing with Words at the National Library of Sweden – Making a Swedish BERT. In *arXiv*, <https://arxiv.org/abs/2007.01658>.
- Jihang Mao and Wanli Liu. 2019. Hadoken: a BERT-CRF Model for Medical Document Anonymization. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019)*, pages 720–726.
- Stephane M Meystre, F Jeffrey Friedlin, Brett R South, Shuying Shen, and Matthew H Samore. 2010. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC medical research methodology*, 10(1):70.
- Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. <http://arxiv.org/abs/1906.05474> Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. In *arXiv*, <https://arxiv.org/abs/1906.05474>.
- Nils Reimers and Iryna Gurevych. 2017. Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks. *arXiv preprint arXiv:1707.06799*.
- Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. 2015. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/UTHealth shared task Track 1. *Journal of Biomedical Informatics*, 58:S11–S19.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Sumithra Velupillai, Hercules Dalianis, Martin Hassel, and Gunnar H Nilsson. 2009. Developing a standard for de-identifying electronic patient records written in Swedish: precision, recall and F-measure in a manual and computerized annotation trial. *International Journal of Medical Informatics*, 78(12):e19–e26.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. <http://arxiv.org/abs/2007.01658> Hugging-Face’s Transformers: State-of-the-art Natural Language Processing. In *arXiv*, <https://arxiv.org/abs/1910.03771>.

An Unsupervised method for OCR Post-Correction and Spelling Normalisation for Finnish

Quan Duong,[♣] Mika Hämäläinen,^{♣,◇} Simon Hengchen[♠]

firstname.lastname@{helsinki.fi;gu.se}

[♣]University of Helsinki, [◇]Rootroo Ltd, [♠]University of Gothenburg

Abstract

Historical corpora are known to contain errors introduced by OCR (optical character recognition) methods used in the digitization process, often said to be degrading the performance of NLP systems. Correcting these errors manually is a time-consuming process and a great part of the automatic approaches have been relying on rules or supervised machine learning. We build on previous work on fully automatic unsupervised extraction of parallel data to train a character-based sequence-to-sequence NMT (neural machine translation) model to conduct OCR error correction designed for English, and adapt it to Finnish by proposing solutions that take the rich morphology of the language into account. Our new method shows increased performance while remaining fully unsupervised, with the added benefit of spelling normalisation. The source code and models are available on GitHub¹ and Zenodo².

1 Introduction

Nature language processing (NLP) is arguably tremendously difficult to tackle in Finnish, due to an extremely rich morphology. This difficulty is reinforced by the limited availability of NLP tools for Finnish in general, and perhaps even more so for historical data by the fact that morphology has evolved through time – some older inflections either do not exist anymore, or are hardly used in modern Finnish. As historical data comes with its own challenges, the presence of OCR errors makes

the data even more burdensome to modern NLP methods.

Obviously, this problematic situation is not unique to Finnish. There are several other languages in the world with rich morphologies and relatively poor support for both historical and modern NLP. Such is the case with most of the languages that are related to Finnish like Erzya, Sami and Komi, these Uralic languages are severely endangered but have valuable historical resources in books that are not yet available in a digital format. OCR remains a problem especially for endangered languages (Partanen, 2017), although OCR quality for such languages can be improved by limiting the domain in which the OCR models are trained and used (Partanen and Riebler, 2019).

Automated OCR post-correction is usually modelled as a supervised machine learning problem where a model is trained with parallel data consisting of OCR erroneous text and manually corrected text. However, we want to develop a method that can be used even in contexts where no manually annotated data is available. The most viable recent method for such a task is the one presented by Hämäläinen and Hengchen (2019). However, their model works only on correcting individual words without considering the context in sentences, and as it focuses on English, it completely ignores the issues rising from a rich morphology. Extending their approach, we introduce a self-supervised model to automatically generate parallel data which is learned from the real OCRred text. Later, we train sequence-to-sequence (seq2seq) NMT models on character level with context information to correct OCR errors. The NMT models are based on the Transformer algorithm (Vaswani et al., 2017), whose detailed comparison is demonstrated in this article.

¹Source Code, <https://github.com/ruathudo/post-ocr-correction>

²Trained models, <https://doi.org/10.5281/zenodo.4242890>

2 Related work

As more and more digital humanities (DH) work start to use the large-scale, digitised and OCRed collections made available by national libraries and other digitisation projects, the quality of OCR is a central point for text-based humanities research. Can one trust the output of complex NLP systems, if these are fed with bad OCR? Beyond the common pitfalls inherent to historical data (see Piotrowski (2012) for a very thorough overview), some works have tried to answer the question stated above: Hill and Hengchen (2019) use a subset of 18th-century corpus, ECCO³ as well as its keyed-in counterpart ECCO-TCP to compare the output of common NLP tasks used in DH and conclude that OCR noise does not seem to be a large factor in quantitative analyses. A conclusion similar to previous work by Rodriguez et al. (2012) in the case of NER and to Franzini et al. (2018) for authorship attribution, but in opposition to Mutuvi et al. (2018) who focus on topic modelling for historical newspapers and conclude that OCR does play a role. More recently and still on historical newspapers, van Strien et al. (2020) conclude that while OCR noise does have an impact, its effect widely differs between downstream tasks.

It has become apparent that OCR quality for historical texts has become central for funding bodies and collection-holding institutions alike. Reports such as the one put forward by Smith and Cordell (2019) rise OCR initiatives, while the Library-of-Congress-commissioned report by Cordell (2020) underlines the importance of OCR for cultural heritage collections. These reports echo earlier work by, among others, Tanner et al. (2009) who tackle the digitisation of British newspapers, the EU-wide IMPACT project⁴ that gathers 26 national libraries, or Adesam et al. (2019) who set out to analyse the quality of OCR made available by the Swedish language bank.

OCR post-correction has been tackled in previous work. Specifically for Finnish, Drobac et al. (2017) correct the OCR of newspapers using weighted finite-state methods, accordance with, Silfverberg and Rueter (2015) do the same for Finnish (and Erzya). Most recent approaches rely on the machine translation (MT) of “dirty” text

into “clean” texts. These MT approaches are quickly moving from statistical MT (SMT) – as previously used for historical text normalisation, e.g. the work by Pettersson et al. (2013) – to NMT: Dong and Smith (2018) use a word-level seq2seq NMT approach for OCR post-correction, while Hämäläinen and Hengchen (2019), on which we base our work, mobilised character-level NMT. Very recently, Nguyen et al. (2020) use BERT embeddings to improve an NMT-based OCR post-correction system on English.

3 Experiment

In this section, we describe our methods for automatically generating parallel data that can be used in a character-level NMT model to conduct OCR post-correction. In short, our method requires only a corpus with OCRed text that we want to automatically correct, a word list, a morphological analyzer and any corpus of error free text. Since we focus on Finnish only, it is important to note that such resources exist for many endangered Uralic languages as well as they have extensive XML dictionaries and FSTs available (see (Hämäläinen and Rueter, 2018)) together with a growing number of Universal Dependencies (Nivre et al., 2016) treebanks such as Komi-Zyrian (Lim et al., 2018), Erzya (Rueter and Tyers, 2018), Komi-Permyak (Rueter et al., 2020) and North Sami (Sheyanova and Tyers, 2017).

3.1 Baseline

We design the first experiment based on the previous work (Hämäläinen and Hengchen, 2019), who train a character-level NMT system. Their research indicates that there is a strong semantic relationship between the correct word to its erroneous forms and we can generate OCR error candidates using semantic similarity. To be able to train the NMT model, we need to extract the parallel data of correct words and their OCR errors. Accordingly, we trained the Word2Vec model (Mikolov et al., 2013) on the Historical Newspaper of Finland from 1771 to 1929 using the Gensim library (Řehůřek and Sojka, 2010). After obtaining the Word2Vec model and its trained vocabulary, we extract the parallel data by using the Finnish morphological FST, Omorf (Pirinen, 2015), provided in the UralicNLP library (Hämäläinen, 2019) and – following Hämäläinen and Hengchen (2019) – Levenshtein edit distance

³Eighteenth Century Collections Online, <https://www.gale.com/primary-sources/eighteenth-century-collections-online>

⁴<http://www.impact-project.eu>

(Levenshtein, 1965). The original approach used a lemma list for English for the data extraction, but we use an FST so that we can distinguish morphological forms from OCR errors. Without the FST, different inflectional forms would also be considered to be OCR errors, which is particularly counterproductive with a highly-inflected language.

We build a list of correct Finnish words by lemmatising all words in the Word2Vec model’s vocabulary: if the lemma is present in the Finnish Wiktionary lemma list,⁵ it is considered as correct and saved as such. Next, for each word in this “correct” list, we retrieve the most similar words from the Word2Vec model. Those similar words are checked to see whether they exist in the correct list or not and separated into two different groups of correct words and OCR errors. Notice that not all the words in the error list are the wrong OCR format of the given correct word, and thus need to be filtered out. Following Hämäläinen and Hengchen (2019), we calculate the Levenshtein edit distance scores of the OCR errors to the correct word and empirically set a threshold of 4 as the maximum distance to accept as the true error form of that given word. As a result, for each given correct word, we have a set of similar correct words including the given one and a set of error words. From the two extracted groups, we do pairwise mapping to have one error word as training input and one correct word as the target output. Finally, the parallel data is converted into a character level format before feeding it to the NMT model for training. For example: *j o l e e n* → *j o k e e n* (“into a river”) pair has the first word is incorrect and the second one is the right form. We follow Hämäläinen and Hengchen (2019) and use OpenNMT (Klein et al., 2017) with default settings, i.e. bi-directional LSTM with global attention (Luong et al., 2015). We train for 10,000 steps and keep the last checkpoint as a baseline, which will be referred to as “NATAS” in the remainder of this paper.

3.2 Methods

In the following subsections we introduce a different method to create a parallel dataset and apply a new sequence to the sequence model to train the data. The baseline approach presented above might introduce noise when we are unable to confidently know that the error word is mapped cor-

rectly to the given correct word, especially in the case of semantically similar words that have similar lengths. Another limitation of the baseline approach is that NMT model usually requires more variants to achieve better performance – something limited by the vocabulary of the Word2Vec model, which is trained with a frequency threshold so as to provide semantically similar words. To solve these problems we artificially introduce OCR-like errors in a modern corpus, and thus obtain more variants of the training word pairs and less noise in the data. We further specialise our approach by applying the Transformer model with context and non-context words experiments instead of the default OpenNMT algorithms for training. In the next section, we detail our implementation.

3.2.1 Dataset Construction

For the artificial dataset, we use the Yle News corpus⁶ which contains more than 700 thousand articles written in Finnish from 2011 to 2018. All the articles are stored in a text file. Punctuation and characters not present in the Finnish alphabet are removed before tokenisation. After cleaning, we generate an artificial dataset by two different methods: random generator and a trained OCR error generator model.

Random Generator As previously stated, we will use a random generator to sample an OCR error word. In OCR text, an error normally happens when a character is misrecognized or ignored. This behavior causes some characters in the word to be missed, altered or introduced. The wrong characters will take a small ratio in the text. Thus, we design algorithm 1 to produce similar errors in the modern corpus.

For each word in the dataset, we will introduce errors to that word by deleting, replacing and adding characters randomly with a threshold of noise rate 0.07. The valid characters to be changed, added or removed must be in the Finnish alphabet, we do not introduce special characters as errors. The idea is that we select a random character position in the string with a probability smaller than noise rate multiplied with length of the string to restrict the percentage of errors in the word. This mean with the long word (eg. 15 characters), there will be always an error proposed. This process is repeated for each action of deleting, replac-

⁵<https://fi.wiktionary.org/wiki/Wikisanakirja:Etusivu>

⁶<http://urn.fi/urn:nbn:fi:lb-2019030701>

Algorithm 1 Random errors generator

```
1: procedure RANDOMERROR(Word, NoiseRate)
2:   Alphas = "abcdefghijklmnopqrstuvwxyzääö"
3:   for Action in [delete, add, replace] do
4:     generate Rand is a random number between 0 and 1
5:     if Rand < NoiseRate × WordLength then
6:       Select a random character position P in Word
7:       if character P is in Alphas then
8:         Do the Action on P with Alphas
9:       end if
10:    end if
11:  end for
12: end procedure
```

ing, adding, thus a word could either have all kinds of errors or none if the random rate is bigger than threshold. A longer word is likely to have more errors than a shorter one.

Trained Generator Similarly to the random generator, we will modify the correct word into an erroneous form, but with a different approach. Instead of pure randomness, we build a model to better simulate OCR erroneous forms. The hypothesis is that if the artificial errors introduced to words have the same pattern as found in the real OCRed text, it would be more effective when applying the resulting model back to the real dataset. For example, the letter “i” and “l” are more likely to be misrecognized than “i” and “g” by the OCR engine.

To build the error generation model, we use the extracted parallel dataset from the NATAS experiment. However, the source and target for the NMT model are reversed to have correctly spelled words as the input and erroneous words as the output from the training. By trying to predict an OCR erroneous form for a given correct spelling, the model can learn an error pattern that mimics the real OCRed text. OpenNMT uses cross entropy loss by default, which causes an issue when applied to solve this problem. In our experiments, the model eventually predicted an output identical to the source because it is the most optimal way to reduce the loss. If we want to generate different output for the input, there is a need to penalize the model when having the same prediction as the input. To solve the problem, we built a simple RNN translation model with GRU (gated recurrent unit) layers and a custom loss function as shown in Equation 2. The loss function is built

up from cross entropy cost function in Equation 1, where $H = \{h^{(1)}, \dots, h^{(n)}\}$ is a set of predicted outcomes from the model and $T = \{t^{(1)}, \dots, t^{(n)}\}$ is the set of targets. We calculate normal cross entropy of predicted output \hat{Y} and the labels Y for finding an optimal way to mimic the target Y , on the other hand, the inverted cross entropy between \hat{Y} and the inputs X is to punish the model if the outcomes are identical to the inputs.

The model’s encoder and decoder each have one embedding layer with 128 dimensions and one GRU layer of 512 hidden units. The input sequences are encoded to have the source’s context, this context is then passed through the decoder. For each next character of the output, the decoder concatenates the source’s context, hidden context and character’s embedded vector. The merged vectors then are passed through a linear layer to give the prediction. The model is trained by teacher enforcing technique with the rate 0.5. This means for the next input character, we either select the top one from the previous output or use the already known next one from the target label.

3.2.2 Models

Parallelisation and long memorisation are weakness characteristic of RNNs in NMT (Bai et al., 2018). Fortunately, Transformer proved to be much faster (mainly due to the absence of recursion), and since they process sequences as a whole they are shown to “remember” information better through their multi-head attention mechanism and positional embedding (Vaswani et al., 2017). Transformer has been shown to be extremely efficient in various tasks (see e.g. BERT (Devlin et al., 2018)), which is why we apply this model to our problem. Our implementation of the Trans-

$$\text{cross_entropy}(H, T) = -\frac{1}{n} \sum_{i=1}^n t^{(i)} \ln h^{(i)} + (1 - t^{(i)}) \ln(1 - h^{(i)}) \quad (1)$$

$$\text{loss} = \text{cross_entropy}(\hat{Y}, Y) + 1 \div \text{cross_entropy}(\hat{Y}, X) \quad (2)$$

former model is based on (Vaswani et al., 2017) and uses the Pytorch framework.⁷ The model contains 3 encoder and decoder layers, each of which has 8 heads of self-attention. We also implement a learned positional encoding and use Adam (Kingma and Ba, 2014) as the optimizer with a static learning rate of $5 \cdot 10^{-4}$ which gave a better convergence compared to the default value of 0.001 based on our experiment. Following prior work, cross entropy was again used as the loss function.

Our baseline NATAS only has fixed training samples extracted from the Word2Vec model. In this experiment, we design a dynamic data loader which generates new erroneous words for every mini-batch while training, allowing the model to learn from more variants at every iteration. As was mentioned in the introduction, we train contextualized sequence-to-sequence character-based models. Instead of feeding a single error word to the model as the input, we combine it with the context words before and after it in sequence. We only consider the correct form of that error word as the label, and are not predicting the context words. The input includes the error (target) word in the middle and its two sides context make up a window of odd number of words. Hence, a valid window sliding over the corpus must have an odd size, for instance 3, 5, etc. The way we construct the input and gold label is presented as follows:

- The window size of n words is selected. The middle word is considered the target word
- The words on left and right of the target are context words
- The input sequence is converted in proper format, for example with window $n=5$:

```
<sos> l e f t <sep> c o n t
e x t <ctx> f a r g e t <ctx>
r i g h t <sep> c o n t e x t
<eos> <pad>, where:
```

- <sos> indicates the start of a sequence;
- <sep> is the separator for the context words;

- <ctx> separates left and right context with the target;
- <eos> indicates the end of a sequence;
- <pad> indicates the padding if needed for mini-batch training.

Following the previous section, the “target” word is generated by creating artificial errors in two different ways: using random generator, and a trained generator. For instance, the word “target” in the example above is modified to “farget”, and the model is trained to predict the output “target”. The gold label is also formatted in the same format, but without any context words. In this case, the label should have this form: <sos> t a r g e t <eos>. After having the pairs of input and label formatted properly, we feed them into the Transformer model with a batch size of 256 – a balance between the speed and accuracy in our case. In this experiment, we evaluate our model with 3 different window sizes: 1, 3, and 5, with the window size of 1 as a special case: there are no context words, and the input is <sos> f a r g e t <eos>. For every window size we train with two different error generators (Random and Trained), and have thus 6 models in total. These models are named hereafter **TFRandW1**, **TFRandW3**, **TFRandW5**, **TFTrainW1**, **TFTrainW3**, and **TFTrainW5**, where *TF* stands for Transformer, *Rand* is for the random generator, *Train* is for the trained generator and *Wn* for a window of n words. We proceeded with the training until the loss converged. All models converged after around 20 epochs. The losses for the *Train* models are ~ 0.064 and those for *Rand* are slightly lower, with ~ 0.059 .

4 Evaluation

We evaluate all proposed models and the NATAS baseline on the Ground Truth Finnish Fraktur dataset⁸ made available by the National Library of Finland, a collection of 479 journal and newspaper pages from the time period 1836 - 1918 (Kettunen

⁸“OCR Ground Truth Pages (Finnish Fraktur) [v1](4.8 GB)”, available at <https://digi.kansalliskirjasto.fi/opensdata>

⁷<https://pytorch.org/>

et al., 2018). The data format is constructed as a csv table with 471,903 lines of words or characters and there are four columns of ground truth (GT) aligned with the output coming from 3 different OCR methods TESSERACT, OLD and FR11 (Kettunen et al., 2018).

Despite the existence of character-level benchmarks for OCR post-correction (e.g. Drobac et al. (2017)), we elect to evaluate models on the more realistic setting of whole words. We would like to note that Finnish has very long words, and as a result this metric is actually tougher. In the previous section, our models are trained without non-alphabet characters, so all the tokens which have non-alphabet will be removed. We also removed the blank lines which have no result from OCR. After having the ground truth and OCR text cleaned, the number of tokens for each OCR method (TESSERACT, OLD, FR11) are 458,799, 464,543 and 470,905 with accuracies of 88.29%, 75.34% and 79.79% respectively. The OCR words will be used as input data for the evaluation of our post-correction systems. The translation processes apply for each OCR method separately with the input tokens formatted based on the model’s requirement. In NATAS, we used OpenNMT to translate with the default settings. In Transformer models with context, we created a sliding window over the rows of the OCRed text. For the non-context model, we only need a single token for source input. These models do the translation with beam search $k = 3$ and the highest probability sequence is chosen as the output. The result is shown in Table 1.

Models	TESSERACT (88.29)	OLD (75.34)	FR11 (79.79)
NATAS	63.35	61.63	64.95
TFRandW1	69.78	67.33	71.64
TFRandW3	70.02	67.45	71.69
TFRandW5	71.24	68.35	72.56
TFTrainW1	70.22	68.30	72.22
TFTrainW3	71.19	69.25	73.14
TFTrainW5	71.24	69.30	73.21

Table 1: Models accuracy on word level for all three OCR methods (%)

4.1 Error Analysis

From the result in Table 1, we can see all the models could not make any improvement on OCR text. However, there is clearly an advantage of using an

artificial dataset and Transformer model for training, which has a 7 percentage points higher accuracy compared to NATAS. After analyzing the result, we found that there are many interesting cases where the output words are considered as errors when compared to the ground truth directly but they are still correct. The difference is that the ground truth has been corrected by maintaining the historical spelling, but as our model has been trained to correct words to a modern spelling, these forms will appear as incorrect when compared directly with the ground truth. However, our models still corrected many of them right, but just happened to normalize the spelling to modern Finnish at the same time. As examples, the word *lukuvuoden* (“academic year”) is normalized to *lukuvuoden*, and the word *kortt* (“card”) is normalized to *korrti*, which are the correct spellings in modern Finnish. So, the problem here is that many words have acquired a new spelling in modern Finnish but are seen as the wrong result if compared to the ground truth, which affects the real accuracy of our models. In the 19th century Finnish text, the most obvious difference compared to modern Finnish is the variation of w/v , where most of the words containing v are written as w in old text, whereas in modern Finnish w is not used in any regular word. Kettunen and Pääkkönen (2016) showed in their experiments that the number of tokens containing letter w contribute to 3.3% of all tokens and 97.5% of those tokens is missrecognized by FINTWOL – a morphological analyzer. They also tried to replace the w with v and the unrecognized tokens decreased to 30.6%. These numbers are significant which give us an idea to apply it on our results to get a better evaluation. Furthermore, there is another issue for our models when they try to make up the new words which do not exist in Finnish vocabulary. For example the word *samppaajaa* is likely created from the word *samppanjaa* (“of Champagne”) which must be the correct one. To solve these issues, we suggested a fixing pipeline for our result:

1. Check if the words exist in Finnish vocabulary using Omorfi with UralicNLP, if not then keep the OCRed words as the output.
2. Find all words containing letter v , replace by letter w .

After the processing with the strategy above, we get updated results which can be found in Tables 2,

3, and 4.

Models	Post processed accuracy	Error words accuracy	Correct words accuracy
NATAS	74.71	16.54	82.43
TFRandW1	80.49	16.13	89.03
TFRandW3	80.79	16.94	89.26
TFRandW5	81.89	17.02	90.49
TFTrainW1	83.05	17.11	91.79
TFTrainW3	83.96	18.15	92.68
TFTrainW5	84.00	18.02	92.75

Table 2: Models accuracy post-processing for Tesseract (88.29%)

Models	Post processed accuracy	Error words accuracy	Correct words accuracy
NATAS	71.19	30.66	84.45
TFRandW1	75.10	28.14	90.47
TFRandW3	75.40	28.26	90.83
TFRandW5	76.26	28.63	91.85
TFTrainW1	78.19	35.07	92.30
TFTrainW3	79.26	36.03	93.41
TFTrainW5	79.17	35.41	93.50

Table 3: Models accuracy post-processing for OLD (75.34%)

Models	Post processed accuracy	Error words accuracy	Correct words accuracy
NATAS	75.06	36.52	84.81
TFRandW1	79.66	36.04	90.71
TFRandW3	80.06	37.00	90.96
TFRandW5	81.09	38.04	91.99
TFTrainW1	82.39	43.39	92.26
TFTrainW3	83.50	45.17	93.21
TFTrainW5	83.34	44.01	93.30

Table 4: Models accuracy post-processing for FR11 (79.79%)

The results in Tables 2, 3 and 4 show a vast improvement for all models with the accuracy increased by 10-12 percentage points. In Tesseract, where the original OCR already has a very high quality with an accuracy of about 88%, there is no gain for all models. The best model in this case is TFTrainW5 with 84% accuracy. The reason for the models' worse performance is that they intro-

duced more errors on the already correct words by OCR than fixing actual error words. While the ratio of fixing the error words (18.02%) is much higher than the ratio of confounding the correct words (7.25%), however, due to the number of correct words taking a much larger part in the corpus, the overall accuracy is decreased. In the OLD setting with accuracy of about 75%, 5 out of 7 models have successfully improved the accuracy of the original text. The highest number comes to TFTrainW3 which outperforms OLD by 3.92 percentage points, and following closely is TFTrainW5 with an accuracy of 79.17%. In OLD, we see better error words corrected (36.03%) compared to Tesseract. The accuracy of the TFTrainW5 model for the already corrected words is also slightly higher with 93.5% versus Tesseract 92.75%. The last OCR method for evaluation is FR11 (79%), where – just like in OLD – 5 out of 7 models surpass the OCR result. Again, the TFTrainW3 gives the highest number with 3.71 percentage points improvement on the OCRred text. While the TFTrainW3 shows surprisingly good results on fixing the wrong words with 45.17% accuracy, the TFTrainW5 performs slightly better at handling the right words. Common to all our proposed models, the window size of 1 somewhat unsurprisingly performs worse within both the *Rand* and *Train* variants.

5 Conclusion and Future work

In this paper, we have shown that creating and using an artificial error dataset clearly outperforms the NATAS baseline (Hämäläinen and Hengchen, 2019), with a clear advantage for the *Train* over the *Rand* configuration. Another clear conclusion is that a larger context window results in increasing the accuracy of the models. Comparing the new results for all three OCR methods, we see the models are most effective with FR11 when the ratio of fixing wrong words (45.17%) is high enough to overcome the issue of breaking the right words (6.7%). Our methods also work very well on OLD with ability to fix 36.03% of wrong words and handle more than 93% of right words correctly. However, our models are not compelling enough to beat the accuracy achieved by Tesseract, a conclusion we see as further work.

In spite of the effectiveness of the post-correction strategy, it does not guarantee that all the words with *w/v* replaced are correct, nor that

UralicNLP manages to recognize all the existing Finnish words. For example: the wrong OCR word *mcntoistamuotiscn* was fixed to *metoistavuotisen* which is the correct one according to the gold standard, but UralicNLP has filtered it out due to not considering that is the valid Finnish word. This is true, as the first syllable *kol* was dropped out due to a line break in the data, and without the line break, the word would be *kolmetoistavuotisen* ("13 years old"). This means that in the future, we need to develop better strategies more suitable to OCR contexts for telling correct and incorrect words apart.

This implies that in reality the corrected cases can be higher if we don't revert the already normalized *w/v* words. In addition, if there is a better method to ensure a word is valid in Finnish, the result could be improved. Thus, our evaluation provides an overall view of how the Transformer and Trained Error Generator models with context words could improve the post OCR correction notably. Our methods also show that using artificial dataset from a modern corpus is very beneficial to normalize the historical text.

Importantly, we would like to underline that our method does not rely on huge amounts of hand annotated gold data, but can rather be applied for as long as one has access to an OCRed text, a vocabulary list, a morphological FST and error-free data. There are several endangered languages related to Finnish that already have these aforementioned resources in place. In the future, we are interested in trying our method out in those contexts as well.

References

- Yvonne Adesam, Dana Dannélls, and Nina Tahmasebi. 2019. Exploring the quality of the digital historical newspaper archive KubHist. *Proceedings of DHN*.
- Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling.
- Ryan Cordell. 2020. Machine learning and libraries: a report on the state of the field. Technical report, Library of Congress.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding.
- Rui Dong and David Smith. 2018. Multi-input attention for unsupervised OCR correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Senka Drobac, Pekka Sakari Kauppinen, and Bo Kristter Johan Linden. 2017. OCR and post-correction of historical Finnish texts. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*. Linköping University Electronic Press.
- Greta Franzini, Mike Kestemont, Gabriela Rotari, Melina Jander, Jeremi K Ochab, Emily Franzini, Joanna Byszuk, and Jan Rybicki. 2018. Attributing authorship in the noisy digitized correspondence of Jacob and Wilhelm Grimm. *Frontiers in Digital Humanities*, 5:4.
- Mika Härmäläinen and Simon Hengchen. 2019. From the paft to the fiiture: a fully automatic NMT and word embeddings method for OCR post-correction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 431–436.
- Mika Härmäläinen and Jack Rueter. 2018. Advances in synchronized xml-media wiki dictionary development in the context of endangered uralic languages. In *Proceedings of the XVIII EURALEX International Congress: Lexicography in Global Contexts*.
- Mika Härmäläinen. 2019. UralicNLP: An NLP library for Uralic languages. *Journal of Open Source Software*, 4(37):1345.
- Mark J Hill and Simon Hengchen. 2019. Quantifying the impact of dirty OCR on historical text analysis: Eighteenth century collections online as a case study. *Digital Scholarship in the Humanities*, 34(4):825–843.
- Kimmo Tapio Kettunen, Jukka Kervinen, and Jani Mika Olavi Koistinen. 2018. Creating and using ground truth ocr sample data for finnish historical newspapers and journals. In *Proceedings of the Digital Humanities in the Nordic Countries 3rd Conference*, CEUR Workshop proceedings, pages 162–169. Technical University of Aachen. Digital Humanities in the Nordic Countries ; Conference date: 07-03-2018 Through 09-03-2018.
- Kimmo Tapio Kettunen and Tuula Anneli Pääkkönen. 2016. Measuring lexical quality of a historical finnish newspaper collection – analysis of garbled ocr data with basic language technology tools and means. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proc. ACL*.

- Vladimir I. Levenshtein. 1965. Двоичные коды с управлением выпадений, вставок и замещений символов. Доклады Академии Наук СССР, 63(4):845–848.
- KyungTae Lim, Niko Partanen, and Thierry Poibeau. 2018. Multilingual dependency parsing for low-resource languages: Case studies on North Saami and Komi-Zyrian. In *Proceedings of LREC 2018*.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Stephen Mutuvi, Antoine Doucet, Moses Odeo, and Adam Jatowt. 2018. Evaluating the impact of OCR errors on topic modeling. In *International Conference on Asian Digital Libraries*, pages 3–14. Springer.
- Thi Tuyet Hai Nguyen, Adam Jatowt, Nhu-Van Nguyen, Mickael Coustaty, and Antoine Doucet. 2020. Neural machine translation with bert for post-ocr error detection and correction. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, pages 333–336.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.
- Niko Partanen. 2017. Challenges in ocr today: Report on experiences from INEL. In *Электронная Письменность Народов Российской Федерации: Опыт, Проблемы И Перспективы*, pages 263–273.
- Niko Partanen and Michael Riebler. 2019. An OCR system for the unified northern alphabet. In *The fifth International Workshop on Computational Linguistics for Uralic Languages*.
- Eva Pettersson, Beáta Megyesi, and Jörg Tiedemann. 2013. An SMT approach to automatic annotation of historical text. In *Proceedings of the workshop on computational historical linguistics at NODALIDA 2013*.
- Michael Piotrowski. 2012. Natural language processing for historical texts. *Synthesis lectures on human language technologies*, 5(2):1–157.
- Tommi A Pirinen. 2015. Development and use of computational morphology of Finnish in the open source and open science era: Notes on experiences with Omorfi development. *SKY Journal of Linguistics*, 28:381–393.
- Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50.
- Kepa Joseba Rodriquez, Mike Bryant, Tobias Blanke, and Magdalena Luszczynska. 2012. Comparison of named entity recognition tools for raw OCR text. In *KONVENS*, pages 410–414.
- Jack Rueter, Niko Partanen, and Larisa Ponomareva. 2020. On the questions in developing computational infrastructure for komi-permyak. In *Proceedings of the Sixth International Workshop on Computational Linguistics of Uralic Languages*, pages 15–25.
- Jack Rueter and Francis Tyers. 2018. Towards an Open-Source Universal-Dependency Treebank for Erzya. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*.
- Mariya Sheyanova and Francis M. Tyers. 2017. Annotation schemes in North Sámi dependency parsing. In *Proceedings of the 3rd International Workshop for Computational Linguistics of Uralic Languages*.
- Miikka Silfverberg and Jack Rueter. 2015. Can morphological analyzers improve the quality of optical character recognition? In *Septentrio Conference Series*, 2, pages 45–56.
- David A. Smith and Ryan Cordell. 2019. A research agenda for historical and multilingual optical character recognition. Technical report, Northeastern University.
- Daniel van Strien, Kaspar Beelen, Mariona Coll Ardanuy, Kasra Hosseini, Barbara McGillivray, and Giovanni Colavizza. 2020. Assessing the impact of ocr quality on downstream nlp tasks. In *ICAART (1)*, pages 484–496.
- Simon Tanner, Trevor Muñoz, and Pich Hemy Ros. 2009. Measuring mass text digitization quality and usefulness. *D-lib Magazine*, 15(7/8):1082–9873.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Learning to Lemmatize in the Word Representation Space

Jarkko Lagus

University of Helsinki
Department of Computer Science
jarkko.lagus@helsinki.fi

Arto Klami

University of Helsinki
Department of Computer Science
arto.klami@helsinki.fi

Abstract

Lemmatization is often used with morphologically rich languages to address issues caused by morphological complexity, performed by grammar-based lemmatizers. We propose an alternative for this, in form of a tool that performs lemmatization in the space of word embeddings. Word embeddings as distributed representations natively encode some information about the relationship between the base and inflected forms, and we show that it is possible to learn a transformation that approximately maps the embeddings of inflected forms to the embeddings of the corresponding lemmas. This facilitates an alternative processing pipeline that replaces traditional lemmatization with the lemmatizing transformation in downstream processing for any application. We demonstrate the method in the Finnish language, outperforming traditional lemmatizers in an example task of document similarity comparison, but the approach is language independent and can be trained for new languages with mild requirements.

1 Introduction

Morphologically rich languages (MRLs) encode more information (such as case, gender, and tense) into single word units, compared to analytical languages like English. For example, Finnish has 15 different word cases for nouns and adjectives. The different cases generate new words from the syntactical point of view, and in combination with plural forms Finnish ends up having 30 different word forms for each noun and adjective.

A rich morphology results in extremely large vocabulary and hence low frequency for most word forms in corpora of reasonable size, causing

problems, e.g., when learning distributed representations – word embeddings – today widely used in most language processing tasks. While embeddings can be trained for MRLs using the traditional methods, such as `fastText` (Bojanowski et al., 2016), `Word2Vec` (Mikolov et al., 2013) and `GloVe` (Pennington et al., 2014), their quality still leaves a lot to desire. For example, the results on standard word embedding tests are often worse for MRLs (Cotterell et al., 2018).

The natural solution for addressing morphological complexity is lemmatizing, often used as pre-processing before analysis. Even though lemmatization loses information by completely ignoring the case, it typically improves performance in various language processing tasks. Transformers and other flexible language models (Devlin et al., 2019; Brown et al., 2020), as well as advanced tokenization methods (Schuster and Nakajima, 2012; Kudo and Richardson, 2018), may have reduced the need for lemmatization in general, but it still remains vital for MRLs for many tasks (Ebert et al., 2016; Cotterell et al., 2018; Kutuzov and Kuzmenko, 2019).

Traditional lemmatization does not, however, resolve all issues caused by rich morphology, especially as part of a pipeline that uses word embeddings. The embeddings themselves are difficult to estimate for MRLs and the embedding methods are typically not transparent about their uncertainty. For instance, the lemma itself may be rare in a typical training corpus and hence we may even switch to using a less reliable embedding, without knowing it. Ebert et al. (2016) proposed a possible resolution of training the embeddings on a lemmatized corpus, but this prevents the use of high-quality pretrained embeddings available for many languages and may otherwise hurt embedding quality. The standard processing pipeline also requires access to a good lemmatizer, which may not be available for rare languages, and some-

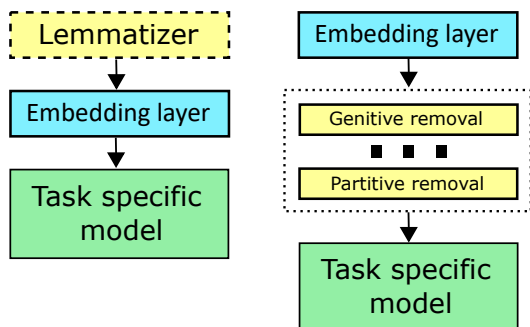


Figure 1: (Left): Traditional task models are trained on the embeddings of either all word forms or the lemmas, obtained by preprocessing with a lemmatizer. (Right): We use embeddings of all word forms but normalize them in the *embedding space*, integrating naturally into the task model.

times they do not work ideally for specialized vocabularies (e.g. medical language). The creation of such a lemmatizer often requires expert knowledge of the target language.

We propose a novel approach for addressing rich morphology, illustrated in Figure 1. Instead of using a traditional lemmatizer to find the lemmas and using the embeddings for those to represent the content, we do the opposite: We start with the embeddings for all original word forms and then perform lemmatization in the embedding space. This is carried out by a neural network that approximately maps the embeddings of inflected forms into the embeddings of the lemmas. We believe that this may provide embeddings that are better for downstream processing tasks compared to the ones available for the lemmas, for instance when the lemma itself is rare since the model is implicitly able to leverage information across multiple words and cases. Another advantage of lemmatization in the embedding space is easy integration as part of the standard modeling workflow that often builds on neural networks anyway, instead of requiring a separate lemmatizer.

Traditional lemmatization is basically a character-level operation, where grammar rules are used to backtrack the basic form that could have generated the inflected form. We, however, consider word inflections as "bias" in the embedding space, so that the embedding for the inflected word combines (in some unknown way) the semantic meaning of the word and the case information. Consequently, our formulation

resembles conceptually the problem of bias removal widely studied in the word embedding literature (Bolukbasi et al., 2016; Brunet et al., 2019). The task in bias removal is to transform the embeddings of individual words such that unwanted systematic biases related to gender etc. disappear. Our approach can be interpreted in this context as a method of removing undesired morphological information while retaining the semantic meaning of the word.

We demonstrate the approach on the Finnish language, restricting the analysis for nouns and adjectives that often contain the most important content words for tasks like document similarity comparison or information retrieval. We use pretrained `fastText` embeddings (Bojanowski et al., 2016) that use subword-level information to provide embeddings for all possible word forms and train a model for mapping them for embeddings of the lemmas using on the dataset extracted from Wiktionary by Durrett and DeNero (2013). The approach is, however, directly applicable to other word classes and languages. Besides the pretrained embeddings, it requires only access to (a) existing list of pairs of lemmas and inflected words as in our case, (b) dictionary and morphological generator, or (c) existing traditional lemmatizer for the language. For instance, `fastText` provides such embeddings for 157 languages, and morphological analyzers or generators exist for most of these.

Besides the core concept of lemmatizing in the embedding space, our main contributions are in the specification of practical details for learning the lemmatizers. We specify four alternative neural network architectures, define a suitable objective function and quality metric, and propose a novel idempotency regularization technique to prevent the models from doing anything else besides the lemmatization. We evaluate the approach in document comparison, outperforming the standard pipeline using traditional lemmatizers, and demonstrate it additionally in the task of word list generation.

An open-source implementation of the method in Python is made available at <https://github.com/jalagus/embedding-level-lemmatization>.

2 Related Work

Even though we are the first to directly consider the task of transforming embeddings to lemmatize words, the general question of addressing rich morphology in distributed representations has been studied from various perspectives.

Cotterell et al. (2018) studied the effect of morphological complexity for task performance over multiple languages. They showed that morphological complexity correlates with poor performance but that lemmatization helps to cope with the complexity. Kutuzov and Kuzmenko (2019) showed a similar effect to hold even with more complex language models, at least for the Russian language. Ebert et al. (2016), in turn, showed that for MRLs we can improve word similarity comparisons by learning Word2Vec embeddings from a lemmatized corpus, rather than training them on all data and lemmatizing while learning the task model.

Kondratyuk et al. (2018) studied supervised lemmatization and morphological tagging using bidirectional RNNs with character and word-level embeddings in MRLs. They showed that a combination of lemma information and morphological tags improve lemmatization and tagging, but may hurt for English. Along similar lines, Rosa and Žabokrtský (2019) suggested using word-embedding clustering to improve lemmatization.

As we consider lemmatization from the perspective of bias removal, our work relates to methods for removal of gender bias (Bolukbasi et al., 2016; Zhao et al., 2017). In this line of work, the embedding space is assumed to encode gender information in specific dimensions, so that bias can be minimized by removing them. The main difference to our work is that their goal is primarily in removing the bias, whereas we look for embeddings that retain the semantic meaning of the word well and that are good for downstream task performance.

3 Evaluation of Lemmatization in Embedding Spaces

A traditional lemmatizer either returns the true lemma or not, but when operating in the embedding space of continuous vector representations the question of correctness needs more attention. We start by discussing the evaluation before proceeding to explain the approach itself that builds on these insights.

First of all, we note that we can use task performance in any downstream task to evaluate the quality – our ultimate goal is in solving the task well, not in learning the embeddings. We will demonstrate this later in the task of document similarity comparison. However, it is highly useful to also have a generic task-independent metric directly measuring the lemmatization accuracy, which can also be used for motivating the objective for training. We want a good word embedding space lemmatizer $M(e_w)$ to simultaneously satisfy two different criteria:

1. Ability to transform any embedding e_w to the embedding of its lemma w' , and
2. Ability to retain embeddings of lemmas or lemmatized embeddings as is.

The first criterion is intuitive, matching our goal, but we need to decide how to measure the similarity. For high-dimensional spaces, it is not reasonable to expect a perfect recovery of the embedding e_w itself, but instead, we should count all embeddings that are close enough as correct. To determine 'close enough', we use a simple definition based on neighborhoods: Lemmatization is correct if the *closest neighbor* for the transformed embedding of a word w is the embedding of its lemma w' . We denote by ACC_{LEM} the accuracy of nearest-neighbor (rank-1) retrieval accuracy for w' in the neighborhood of $M(e_w)$, using Euclidean distance for similarity.

The second criterion is imposed as we want to consider the lemmatization step as a black box for which we can feed in arbitrary words, including those that are already lemmas. The lemmatizer should not alter them in any way. We measure this by an indirect metric of ACC_{IDEM} , which corresponds to the rank-1 retrieval accuracy for w' in the neighborhood of $M(M(e_w))$, the output for an embedding e_w passed *twice* through the model. For a more detailed discussion and justification, see Section 4.3.

Together the metrics ACC_{LEM} and ACC_{IDEM} characterize the general ability of any embedding-space lemmatizer in a model-independent way; both are based on retrieval accuracy and can be evaluated without additional assumptions besides the distance measure. We will later use them also to motivate our objective function, a differentiable approximation for their weighted combination.

4 Approach

Denoting an arbitrary inflected form word embedding by e_w and the related lemma word embedding by $e_{w'}$, we wish to learn some mapping $M(\cdot)$ such that $M(e_w|\theta) \approx e_{w'}$. We do this by assuming a parametric model family, a neural network, and learning its parameters θ based on a collection of $(e_w, e_{w'})$ pairs of pretrained embeddings in a supervised fashion. For simplicity of notation, we omit the parameters and simply write $M(e_w)$ instead of $M(e_w|\theta)$ for the rest of the paper.

We hypothesize that inflected forms lie on a specific subspace of the embedding space (see Figure 2) and that we can retrieve the lemmatized forms by a simple, but a possibly nonlinear, transformation in the embedding space. This can be interpreted as the removal of "bias" caused by the inflection. We want this mapping to be lightweight so that it can be integrated as part of a task model with a small computational overhead. Complex transformations are discouraged also because they would increase the risk of altering the semantic content captured by the embedding.

We discuss two alternative ways of lemmatizing in the embedding space. The first approach learns a separate model $M_c(e_w)$ for each word case c so that e.g. partitives and genitives are processed with different models. This allows using simple models even if all of the rich morphology was not constrained in low-dimensional subspaces, and also allows reversing the model for morphological generation (see Section 7).

For the processing of arbitrary words with an unknown case, we can make a function composite of multiple models, so that the output of one model is always fed as input for the next one. For instance, to lemmatize both partitives and genitives we can compute $(M_p \circ M_g)(e_w) = M_{partitive}(M_{genitive}(e_w))$, in either order. Assuming the models do nothing else besides remove the effect of the particular case, then this composite function performs the same operation as either model alone, depending on the case of the input word. We naturally cannot guarantee the transformations work exactly like this, but will later present a regularization technique that specifically encourages the models to focus only on the case removal and show empirically that such function composition of multiple models works well.

The other alternative is learning a single global model $M(e_w)$ that can lemmatize all word forms.

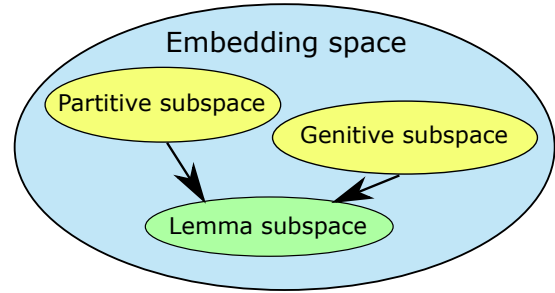


Figure 2: Embedding space as a union of inflected subspaces. Each word class creates a subspace and arrows represents the mappings we wish to learn in order to do lemmatization in the embedding space.

We demonstrate also this approach, but our main focus is on the separate models for each case.

4.1 Neural Architectures

We use feedforward neural networks as models $M_c(e_w)$, restricting the architecture choice for small networks to retain computational efficiency. Both input and output dimensionality needs to match the dimensionality of the embedding, in our case $d = 300$. We investigate empirically four alternative architectures:

1. **Linear**

$$W_1 e_w + b_1, \text{ where } W_1 \in R^{d \times d}$$

2. **Simple**

$$W_2 R(W_1 e_w + b_1) + b_2, \text{ where } W_1 \in R^{500 \times d}, \\ W_2 \in R^{d \times 500}$$

3. **Compression**

$$W_2 R(W_1 e_w + b_1) + b_2, \text{ where } W_1 \in R^{100 \times d}, \\ W_2 \in R^{d \times 100}$$

4. **Complex**

$$W_3 (R(W_2 R(W_1 e_w + b_1) + b_2) + b_3), \text{ where } \\ W_1 \in R^{500 \times d}, W_2 \in R^{500 \times 500}, W_3 \in \\ R^{d \times 500}$$

In all variants, $R(\cdot)$ denotes the rectified linear unit and b_i is a bias term of proper size.

The linear model is motivated by the property of some embeddings encoding various properties as linear relationships (e.g. *king - man + woman ≈ queen*) and fast computation. However, there are no guarantees a linear transformation is sufficient for lemmatization and hence we consider also the three simple nonlinear architectures with at most two hidden layers. Other architectures could certainly be used and a more careful choice

of a specific architecture could further improve the lemmatization accuracy, but we will later show that already these lightweight models work well in practice.

4.2 Objective and Training

To learn models such that $M_c(e_w) \approx e_{w'}$ we need to optimize for a loss function that penalizes for difference between $M_c(e_w)$ and $e_{w'}$ for known pairs of w and w' . As explained in Section 3, we will eventually measure the quality by nearest-neighbor retrieval in the embedding space. Directly optimizing for that is difficult, and hence we optimize for a natural proxy instead, minimizing the squared Euclidean distance

$$D(M_c(e_w), e_{w'}) = \|M_c(e_w) - e_{w'}\|^2.$$

Note that often the norm of the embeddings is considered irrelevant and consequently e.g. Word2Vec (Mikolov et al., 2013) used cosine similarity to measure distances. We want to retain the norms that for some embeddings encode information about e.g. word frequency (Schakel and Wilson, 2015) and hence chose the Euclidean distance.

For training the model we need a collection of N pairs of embeddings for words w and their lemmas w' . Assuming an embedding library that provides embeddings for large vocabulary (or even arbitrary word forms, building on subword-level embeddings (Bojanowski et al., 2016)) we simply need some way of constructing these pairs. The two practical alternatives for this are

- Dictionary of lemmas w' and a morphological generator to form w_c for cases c
- Collection of words w and a traditional lemmatizer for obtaining their lemmas w'

For case-specific models we only use pairs corresponding to the case, whereas for the global model we can pool all pairs, potentially having multiple cases for the same lemma in the training data.

4.3 Idempotency Regularization

Any model trained as above learns to map w to w' , but we cannot tell what it does for words that are already lemmas or that belong to some other case if training a case-specific model. One could in principle add pairs of (w', w') into the training set to address the former, but to prevent transforming

words of other classes we would need similar pairs for *all* possible cases. This would be extremely inefficient.

To avoid transforming the embeddings of other word forms, we propose an alternative of novel regularization strategy encouraging *idempotency*, meaning that the same transformation applied multiple times will not change the output beyond the initial result. We do this by measuring the Euclidean distance $D(M_c(e_w), M_c(M_c(e_w)))$ between the output of the model $M_c(e_w)$ (the supposed lemmatized embedding) and the result of passing the input through the model twice, $M_c(M_c(e_w))$. By encouraging this distance to be small we encourage the model to only remove the information about the particular case, without otherwise changing the embedding. Conceptually this is related to regularization techniques like Barone et al. (2017) designed to prevent catastrophic forgetting (Kirkpatrick et al., 2017); both prevent losing the already learned structure while allowing the model to adapt to a new task.

In practice we minimize the objective

$$L(e_w, e_{w'}) = \alpha \times D(M_c(e_w), e_{w'}) + (1 - \alpha) \times D(M_c(e_w), M_c(M_c(e_w))), \quad (1)$$

where $\alpha \in [0, 1]$ controls the amount of regularization. With $\alpha = 1$ we only optimize the loss and by decreasing the parameter we start regularizing the solution using idempotency. Note that the extreme of $\alpha = 0$ is not meaningful, since the loss term disappears.

5 Model Validation

We validate the approach and the modeling choices (architecture and regularization), using morphologically rich Finnish as an example language. We first evaluate the performance in a task-agnostic manner, before demonstrating case examples in the following two sections.

Data We validate the approach on Finnish language, using pretrained embeddings provided by the `fastText` library (Bojanowski et al., 2016). The embeddings were trained on Common Crawl and Wikipedia corpora and have dimensionality of $d = 300$.

The lemmatization models are trained on the data provided by Durrett and DeNero (2013) which contains words extracted from the open dictionary *Wiktionary*. It directly provides pairs of

inflected and base forms for words, so we do not need to construct them. For Finnish, the dataset contains 1,136,492 word pairs of adjectives and nouns both in singular and plural, resulting in roughly 42,000 word pairs per word case. Each row in the dataset is a pair of form (w, w') which are then transformed to pairs of word embeddings $(e_w, e_{w'})$ using the `fastText` library.

Training We use AdamW optimizer (Kingma and Ba, 2014; Loshchilov and Hutter, 2018) with a learning rate of 0.0002 and a batch size of 32 for training the models in all experiments, but all reasonable stochastic optimization algorithms would work. We separately validated in preliminary tests that running the optimization until convergence of the training objective does not result in overfitting, and hence for the rest of the experiments we used 50 epochs for training to make sure the models are fully converged. In practice, 20-30 epochs were always enough. All experiments shown here are efficient, so that training individual models on consumer-grade 8-core CPU was done in the order of minutes.

Model architectures To compare different architectures, we trained individual models $M_c(e_w)$ on all 15 word cases of Finnish with $\alpha = 1.0$ (i.e. no regularization), not separating plural and singular word cases so that always 10,000 word pairs were used for training and 1,000 for testing. The word pairs for training and test sets were chosen randomly. For the final score, we averaged 10 different runs over randomized splits of the data so that the splits were the same for all models for each run.

Table 1 compares the four different architectures in terms of metrics explained in Section 3, presenting the average accuracy over all word cases (the results are consistent over different cases, not shown here). The main result is that except for the *compression* architecture the accuracies ACC_{LEM} are very similar. This suggests there may not be a specific low-dimensional subspace that is sufficient for lemmatization, but that it can be modeled with fairly simple architectures nevertheless. In terms of ACC_{IDEM} , all models here coincidentally converge to the same value that is close to perfect despite not regularizing for idempotency.

We also trained a global model $M(e_w)$ for lemmatizing all cases using the *simple* model architec-

Model	ACC_{LEM}	ACC_{IDEM}	Time/epoch (s)
linear	0.908	0.978	1.363
compression	0.870	0.978	1.807
simple	0.915	0.978	3.044
complex	0.911	0.978	3.669
global	0.974	0.998	11.145

Table 1: Lemmatization accuracy (ACC_{LEM}) and idempotency criterion (ACC_{IDEM}) for alternative network architectures for case-specific models, averaged over all 15 word cases. The global model can process all cases, but the numerical accuracy is not directly comparable due to a different number of test instances.

ture, using a combined data set of 50,000 examples covering the different cases and 5,000 word pairs for evaluation. Note, however, that the evaluation set was not the same as for the case-specific models that all used only pairs for the specific case. Hence the numbers in Table 1 are not directly comparable, but we can still confirm that also the global model learns to lemmatize well.

Function composition and idempotency regularization When training separate models for each word case c , we need a function composition of multiple models in order to process arbitrary input word forms. To perform this, we need idempotency regularization to prevent individual models from transforming words of wrong cases.

Table 2 demonstrates the effect of the regularization parameter α for an example sentence, using two models trained for lemmatizing genitives and partitives and their combination as function composition. For very small α already the individual models fail due to almost ignoring the main task, whereas for very large α (no regularization) the composition breaks. With $\alpha = 0.4$ we can accurately lemmatize both forms.

6 Application: Document Comparison

To demonstrate the method in a typical application, we consider the task of document comparison where the lemmas of content words often provide sufficient information on similarity. We use a dataset provided by the Finnish national broadcasting company *Yle*¹ containing news articles written in easy-to-read Finnish. We created an artificial dataset by splitting news articles into

¹<http://urn.fi/urn:nbn:fi:lb-2019121205>

α	Word case	Example sentence								
-	original	Leijona oli saanut	paitsi	hyvän ja	nöyrän mielen	myös	monta	uutta	ystävää	
0.1	genitive	Leijona oli saanut	näinen	hyvä	pipopää	nöyrä	tahdonvoima	näinen	iso	uusi ystävää
0.1	partitive	Leijona oli saanut	paitsi	hyvä	pehmyt	nöyrä	mielen	myös	muutama	uusi ystävä
0.1	gen + part	Leijona oli saanut	paitsi	hyvä	pipopää	nöyrä	tunteellisuus	näinen	pieni	uusi tyttökaveri
0.4	genitive	Leijona oli saanut	paitsi	hyvä ja	nöyrä	mielen	myös	monta	uutta	ystävää
0.4	partitive	Leijona oli saanut	paitsi	hyvän ja	nöyrän	mielen	myös	monta	uusi	ystävä
0.4	gen + part	Leijona oli saanut	paitsi	hyvä ja	nöyrä	mielen	myös	monta	uusi	ystävä
0.7	genitive	Leijona oli saanut	paitsi	hyvä ja	nöyrä	mielen	myös	monta	uutta	ystävää
0.7	partitive	Leijona oli saanut	paitsi	hyvän ja	nöyrän	mielen	myös	useampi	uusi	ystävä
0.7	gen + part	Leijona oli saanut	paitsi	hyvä ja	nöyrä	mielen	myös	monta	uusi	ystävä
1.0	genitive	Leijona oli saanut	paitsi	hyvä ja	nöyrä	mielen	myös	monta	uutta	ystävää
1.0	partitive	Leijona oli saanut	paitsi	hyvän ja	nöyrän	mielen	myös	monta	uusi	ystävä
1.0	gen + part	Leijona oli saanut	paitsi	hyvä yskäkin	nöyrä	tahdonvoima	myös	muutama	uusi	ystävä
0.3	global	Leijona oli saanut	paitsi	hyvä ja	nöyrä	mielen	myös	monta	uusi	ystävä
-	ground truth	Leijona oli saanut	paitsi	hyvä ja	nöyrä	mieli	myös	monta	uusi	ystävä

Table 2: Idempotency regularization for function composition of separate models for lemmatizing genitives and partitives. Both too large and small α introduce mistakes for this example sentence, but with $\alpha = 0.4$ and the alternative of global model the result is near perfect. The words in genitive form in the original sentence are $\{hyvän, nöyrän, mielen\}$, and the words in partitive form are $\{uutta, ystävää\}$

two halves and try to predict which two parts belong together by ranking the articles via average vector document representations. We take only a subset of the data, using the first 10,000 news articles from the first three months of the year 2018.

We compare the proposed approach against a conventional pipeline that first lemmatizes the words using the uralicNLP library (Hämäläinen, 2019) (and then uses embeddings for the lemmas for the task) and a pipeline that directly uses the embeddings for all word forms. For the proposed approach we perform lemmatization in the embedding space for four different cases and their combinations, using the *simple* architecture.

For all methods, we form a representation for the document by computing the mean of the word embeddings for all words in the document and use cosine similarity between these mean embeddings to compare documents. One could alternatively consider richer document representations (Wieting et al., 2015; Arora et al., 2017; Gupta et al., 2020) or more accurate similarity metrics (Torki, 2018; Lagus et al., 2019) that might improve the overall accuracy, but we chose the most commonly used approach that is easy to understand to focus on demonstrating the effect of the lemmatization.

We measure performance by retrieval accuracy, by computing the rank of the second half of a given document amongst the set of all 10,000 second halves. Figure 3 shows the overall performance of the different model variants as a func-

tion of the regularization parameter, measured by rank-1 accuracy. We observe three clear results: (a) all ways of lemmatization clearly improve the task performance compared to no lemmatization at all, (b) lemmatization in the embedding space using case-specific models is considerably better than the alternatives of traditional lemmatizer and the global model lemmatizing in the embedding space, and (c) idempotency regularization is crucial, but the method is extremely robust with respect to the specific choice of α – all values between 0.2 and 0.9 result in almost identical performance.

Table 3 illustrates the task performance in more detail for models trained using good choices for the regularization parameter α , measured using retrieval accuracy with different ranks. The results are consistent over the ranks, with case-specific lemmatizers in the embedding space consistently outperforming the other methods.

7 Application: Word List Generation

Even though our main goal is to learn lemmatizers, we note that that the approach is more general. Instead of training a lemmatizer $M_c(e_w) \approx e_{w'}$, we can use the exact same architectures and data for training $G_c(e_{w'}) \approx e_w$ to learn *generators* that provide the embedding for the inflected form for some particular case c .

We demonstrate this via the simple application of word list expansion, which could be used simi-

Model	Word case	α	R@1	R@2	R@3	R@4	R@5	R@6	R@7	R@8	R@9	R@10
simple	gen	0.8	0.368	0.455	0.502	0.534	0.560	0.582	0.599	0.613	0.625	0.636
simple	gen + part	0.8	0.390	0.483	0.533	0.569	0.594	0.613	0.632	0.647	0.658	0.669
simple	gen + ine + part	0.5	0.395	0.491	0.540	0.573	0.597	0.620	0.636	0.649	0.663	0.674
simple	gen + ine + cla + part	0.5	0.390	0.482	0.533	0.567	0.594	0.614	0.631	0.645	0.657	0.668
global	-	0.9	0.329	0.411	0.458	0.488	0.513	0.532	0.548	0.561	0.573	0.585
lemmatizer	-	-	0.311	0.391	0.434	0.464	0.485	0.503	0.518	0.532	0.543	0.552
none	-	-	0.286	0.362	0.404	0.431	0.454	0.474	0.490	0.501	0.514	0.526

Table 3: The best combinations of each model version averaged over 10 different subsets of the news data. R@K means that we rank the documents by similarity and measure the accuracy of the relevant document being within the top K documents.

larly to query expansion for retrieval tasks. Given a list of words w' provided in base form and their embeddings $e_{w'}$, we form a list of embeddings for different inflected forms. We trained case-specific models $G_c(e_{w'})$ similar to before with different values for α , observing a similar trend: the method is robust for the choice, as long as extreme values are avoided.

Table 4 illustrates the method for the word list $\{\text{jääkiekko, Suomi, Venäjä}\}$ ($\{\text{ice hockey, Finland, Russia}\}$ in English) one could use as keywords for searching information about ice hockey matches between the two countries. We show here the words with the embeddings closest to the ones provided by the generator models to verify it works as intended, but in real use, we would naturally use the transformed embeddings directly for the retrieval task – they are likely to be better representations especially for rare cases for which the actual pre-computed embedding e_w is likely to be noisy.

8 Conclusions

For MRLs lemmatization helps in many tasks. We showed that the conventional pipeline using traditional lemmatizers as preprocessing can be replaced by lemmatization in the embedding space. Already simple neural networks can transform the embeddings of inflected words so that the closest word in the embedding space is of the correct lemma. This verifies lemmatization in the embedding space is possible, but in real applications, we naturally would not convert the result back to the lemma. Instead, any downstream task simply processes the lemmatized embeddings directly.

We showed that the method outperforms conventional lemmatization preprocessing in the document similarity comparison task, which implies we are not merely learning to replicate the exact lemmatization but instead learn embeddings that

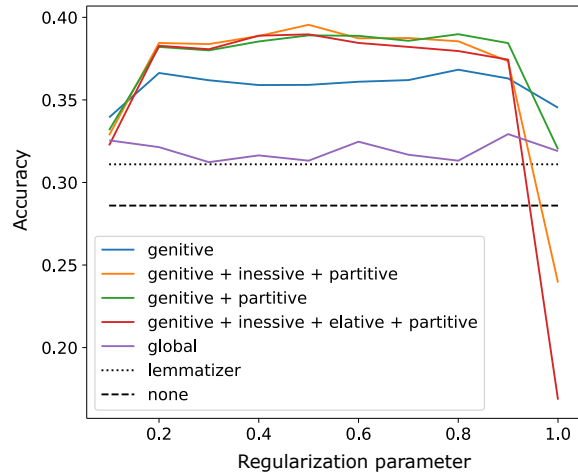


Figure 3: The effect of the regularization parameter (the α parameter) on full-length document comparison task using rank-1 accuracy as the scoring method. There is a notable improvement over the baselines (lemmatizer and none) when using our models with idempotency regularization parameter chosen within the range (0.2, 0.9), and the improvement is highly insensitive to the specific value of the parameter.

better capture the word content. We hypothesize this is related to how rare words are represented in the embedding space; for rare words, the embeddings for all word forms are unreliable, including the one for the lemma itself. Subword-level embeddings, like `fastText` used in our experiments, may still be able to learn sensible embeddings for the collection of all inflected forms together, and by lemmatizing in the embedding space we borrow some information from all of the forms. In other words, we argue that the approximate lemmatization performed by the neural network may have the regularizing ability to reduce noise in embeddings of rare words so that the 'approximation' is actually better than the target embedding used during training.

Word case	Expanded form	Ground truth
genitive	jääkiekon	jääkiekon
genitive	Suomen	Suomen
genitive	Venäjän	Venäjän
inessive	<i>jääkiekkossa</i>	jääkiekossa
inessive	Suomessa	Suomessa
inessive	Venäjässä	Venäjässä
elative	jääkiekosta	jääkiekosta
elative	Suomesta	Suomesta
elative	Venäjältä	Venäjältä
partitive	jääkiekkoa	jääkiekkoa
partitive	Suomea	Suomea
partitive	Venäjää	Venäjää
illative	jääkiekkoon	jääkiekkoon
illative	Suomeen	Suomeen
illative	Venäjälle	Venäjälle

Table 4: Example word list expansion generated for the word list $\{\text{jääkiekko}, \text{Suomi}, \text{Venäjä}\}$ ($\{\text{ice hockey}, \text{Finland}, \text{Russia}\}$) using morphological generator models for genitive, inessive, elative, partitive, and illative cases with regularization parameter $\alpha = 0.4$. Note the mistake for the inessive case of "jääkiekko", which should be "jääkiekossa" and not "jääkiekkossa" – the word has the correct "-ssa" suffix but the root is incorrect. It is also worth noting that "jääkiekkossa" is not a valid word form in Finnish at all, but the `fastText` library provides embeddings for arbitrary strings using sub-word information. The embeddings for the two forms are likely very close, and hence the mistake would have no effect in retrieval tasks.

In this work we presented the overall concept for lemmatization in the embedding space and experimented on various technical choices, building the basis for future development. Our main findings were that a global model can perform lemmatization well when measured only by accuracy, but for the task of document comparison, we reached considerably better results by function composition of case-specific models. To make this possible we proposed a novel idempotency regularization, and showed that the approach is highly robust for the choice of the regularization parameter, making it essentially parameter-free. Finally, we note that even though we demonstrated the approach for an example MRL language Finnish and only for lemmatization of nouns and adjectives, the method is general and directly applicable for other languages and word classes.

Acknowledgements

This work was supported by the Academy of Finland Flagship programme: Finnish Center for Artificial Intelligence, FCAI.

References

- Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of International Conference on Learning Representations*.
- Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. *arXiv preprint arXiv:1707.09920*.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.
- Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29:4349–4357.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Marc-Etienne Brunet, Colleen Alkalay-Houlihan, Ashton Anderson, and Richard Zemel. 2019. Understanding the origins of bias in word embeddings. In *International Conference on Machine Learning*, pages 803–811. PMLR.
- Ryan Cotterell, Sabrina J. Mielke, Jason Eisner, and Brian Roark. 2018. Are all languages equally hard to language-model? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 536–541, New Orleans, Louisiana. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North*

- American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1185–1195, Atlanta, Georgia. Association for Computational Linguistics.
- Sebastian Ebert, Thomas Müller, and Hinrich Schütze. 2016. Lamb: A good shepherd of morphologically rich languages. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 742–752.
- Vivek Gupta, Ankit Saw, Pegah Nokhiz, Praneeth Netrappalli, Piyush Rai, and Partha Talukdar. 2020. P-sif: Document embeddings using partition averaging. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7863–7870.
- Mika Härmäläinen. 2019. UralicNLP: An NLP library for Uralic languages. *Journal of Open Source Software*, 4(37):1345.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.
- Daniel Kondratyuk, Tomáš Gavenčík, Milan Straka, and Jan Hajič. 2018. LemmaTag: Jointly tagging and lemmatizing for morphologically rich languages with BRNNs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4921–4928, Brussels, Belgium. Association for Computational Linguistics.
- Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.
- Andrey Kutuzov and Elizaveta Kuzmenko. 2019. To lemmatize or not to lemmatize: How word normalisation affects ELMo performance in word sense disambiguation. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 22–28, Turku, Finland. Linköping University Electronic Press.
- Jarkko Lagus, Janne Sinkkonen, Arto Klami, et al. 2019. Low-rank approximations of second-order document representations. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. ACL.
- Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Rudolf Rosa and Zdeněk Žabokrtský. 2019. Unsupervised lemmatization as embeddings-based word clustering. *arXiv preprint arXiv:1908.08528*.
- Adriaan MJ Schakel and Benjamin J Wilson. 2015. Measuring word significance using distributed representations of words. *arXiv preprint arXiv:1508.02297*.
- Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.
- Marwan Torki. 2018. A document descriptor using covariance of word vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 527–532.
- John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.
- Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989, Copenhagen, Denmark. Association for Computational Linguistics.

Synonym Replacement based on a Study of Basic-level Nouns in Swedish Texts of Different Complexity

Evelina Rennes, Arne Jönsson

Department of Computer and Information Science

Linköping University, Linköping, Sweden

evalina.rennes@liu.se, arne.jonsson@liu.se

Abstract

In this article, we explore the use of basic-level nouns in texts of different complexity, and hypothesise that hypernyms with characteristics of basic-level words could be useful for the task of lexical simplification. Basic-level terms have been described as the most important to human categorisation. They are the earliest emerging words in children's language acquisition, and seem to be more frequently occurring in language in general. We conducted two corpus studies using four different corpora, two corpora of standard Swedish and two corpora of simple Swedish, and explored whether corpora of simple texts contain a higher proportion of basic-level nouns than corpora of standard Swedish. Based on insights from the corpus studies, we developed a novel algorithm for choosing the best synonym by rewarding high relative frequencies and monolexemy, and restricting the climb in the word hierarchy not to suggest synonyms of a too high level of inclusiveness.

1 Introduction

The research concerned with automatically reducing the complexity of texts is called *Automatic Text Simplification* (ATS). Automatic text simplification was first proposed as a pre-processing step prior to other natural language processing tasks, such as machine translation or text summarisation. The assumption was that a simpler syntactic structure would lead to less ambiguity and, by extension, a higher quality of text processing (Chandrasekar et al., 1996). However, one of the main goals of modern automatic text simplification systems is to aid different types of target readers. The

manual production of simple text is costly and if this process could be automated, this would have a beneficial effect on the targeted reader, as well as the society as a whole. Previous ATS studies have targeted different reader groups, such as second language (L2) learners (Petersen and Ostendorf, 2007; Paetzold, 2016), children (De Belder and Moens, 2010; Barlacchi and Tonelli, 2013; Hmida et al., 2018), persons with aphasia (Carroll et al., 1998; Canning and Tait, 1999; Devlin and Unthank, 2006), the hearing-impaired (Inui et al., 2003; Daelemans et al., 2004; Chung et al., 2013), and other persons with low literacy skills (Alufisio et al., 2008; Candido Jr et al., 2009; Alufisio et al., 2010). Reducing the complexity of a text can be done in numerous ways but one of the subtasks of ATS is lexical simplification: the process of finding and replacing difficult words or phrases with simpler options. Finding such simpler words can be done by using frequency measures to choose between substitution candidates with the intuition that the more common a word is, the simpler a synonym it is. As pointed out, for instance by Alfter (2021), more frequent words can also be complex as they tend to be more polysemous.

Finding simpler words can also be done by studying how human writers do. To write simple texts, the writers usually consult guidelines. For Swedish, such guidelines are given by Myn-digheten för Tillgängliga Medier (MTM)¹. The MTM guidelines state, among other things, that the text should be adapted to the type of reader who is going to read the text, and that everyday words should be used (MTM, 2020).

In this article, we explore the use of basic-level nouns in texts of different complexity, and hypothesise that hypernyms with characteristics of basic-level words could be useful for the task of lexical simplification. We then use this knowledge to cre-

¹Swedish Agency for Accessible Media

ate an algorithm for synonym replacement. The conventional definition of a *synonym* is a word that have the same or nearly the same meaning as another word. However, for simplicity, in this article we extend this notion to also include near-synonyms or other semantically similar words.

Hypernyms have been previously studied from the perspective of lexical simplification. For example, Drndarević and Saggion (2012) explored the types of lexical simplification operations that were present in a parallel corpus comprising 200 standard and simple news texts in Spanish, and found that the exchanged words could be hypernyms, hyponyms and meronyms. Biran et al. (2011) used the vocabularies of Wikipedia and Simple English Wikipedia to create word pairs of content words, and one of the methods for filtering out substitution word pairs was to consult the synonym and hypernym relations between the words. Comparable synonym resources for Swedish include SynLex (Kann and Rosell, 2005) and SweSaurus (Borin and Forsberg, 2014).

Given what we know how simple texts are written, it seems probable that a corpus of simple text, targeting children and readers with different kinds of disabilities, is characterised by a higher proportion of basic-level nouns than, for example, a corpus comprising texts that are said to reflect general Swedish language of the 90's. The aim of this study was to explore this claim in corpora of simple and standard texts, and to see how this could be used in the context of lexical text simplification.

2 Basic-level Words

Prototype theory, as defined by Rosch et al. (1976), claims that there is a scale of human categorisation where some representing concepts are more representative than others. For example, *furniture* can be regarded as higher up in the taxonomy than *chair* or *table*, whereas *kitchen chair* or *dining table* can be found at a lower level with higher specificity. Rosch et al. (1976) found that the basic level is the most important to human categorisation. For example, basic-level terms emerge early in a child's language acquisition, and such terms generally seem to be more frequently occurring in language. Another characteristic of basic-level terms is that they often comprise one single lexeme, while subordinate terms more often consist of several lexemes (Evans, 2019).

Theories in cognitive linguistics are important

for computational linguists as they adopt a usage-based approach. This means that language use is essential to how our knowledge of language is gained, and plays a large role in language change and language acquisition (Evans, 2019). When a child learns a language, the knowledge is gathered through extraction of constructions and patterns, a process grounded in general cognitive processes and abilities. One of the central ideas in the usage-based approach is that the relative frequency of linguistic constructions (such as words) affects the language system so that more frequent constructions are better entrenched in the system, thus further influencing language use.

Within the field of cognitive linguistics corpora is one of the proposed methods to study language (Evans, 2019). Corpora make it relatively simple to perform large-scale analyses in order to get quantitative measures on how language is used in a naturalistic setting. The simplest measures we can use are frequency counts, which can provide insights in how commonly used certain constructions are, in comparison with others.

3 Corpus Analysis

We conducted two corpus studies using different corpora.

The first study aimed to compare two corpora, where the first corpus contained texts that reflect the Swedish language, and the second corpus contained easy-to-read texts. The *Stockholm-Umeå Corpus (SUC)* corpus (Ejerhed et al., 2006) is a balanced corpus of Swedish texts written in the 1990's. In this study, we used the 3.0 version of the corpus (*SUC3*).

The *LäSBarT* corpus (Mühlenbock, 2008), is a corpus of Swedish easy-to-read texts of four genres: easy-to-read news texts, fiction, community information, and children's fiction. The *LäSBarT* corpus was compiled in order to mirror simple language use in different domains and genres but it is not truly balanced in the traditional sense.

The hypothesis was that the *SUC3* corpus would exhibit a higher average number of steps to the top-level noun than the *LäSBarT* corpus.

The second study aimed to investigate whether the genre did play a role. In order to investigate this, we conducted an analysis of a corpus of the Swedish newspaper *8 Sidor*, that comprises news articles in Simple Swedish, and a corpus with Göteborgs-Posten articles (*GP2D*). The cor-

pora were of the same genre, but not parallel.

The hypothesis was that the *GP2D* corpus would exhibit an even higher average number of steps to the top-level noun than the *8 Sidor* corpus. The *SUC3* corpus is balanced and, hence, also includes, for instance, simple texts that may affect the difference between the corpora.

3.1 Procedure

All nouns of the resources were extracted, together with their most probable sense gathered from SALDO (Svenskt Associationslexikon) version 2 (Borin et al., 2008). SALDO is a descriptive lexical resource that, among other things includes a semantic lexicon in the form of a lexical-semantic network.

SALDO was also used for extracting lexical relations. For each such noun, we recursively collected all *primary parents* of the input word. The *primary* descriptor describes an entry which better than any other entry fulfils two requirements: (1) it is a semantic neighbour of the entry to be described (meaning that there is a direct semantic relationship, such as synonymy, hyponymy, and meronymy, between words); and (2) it is more central than the given entry. However, there is no requirement that the *primary* descriptor is of the same part of speech as the entry itself.

The number of steps taken to reach the top-level noun was counted. The algorithm ended when there were no more parents tagged as a noun. The method was inspired by the collection of synonym/near-synonym/hypernym relations in Borin and Forsberg (2014).

In addition to this analysis, we also collected the frequency counts of the nouns occurring in the corpora and their superordinate nouns, as well as indication of compositionality. The frequency measures used were relative frequencies gathered from the *WIKIPEDIA-SV* corpus, accessed through Språkbanken².

3.2 Corpus Analysis Results

The number of extracted instances were 206,609 (*SUC3*), 177,390 (*LäsBarT*), 180,012 (*GP2D*), and 543,699 (*8 Sidor*). The distribution of the number of words per superordinate level is presented in Figure 1.

In the first study, we compared the *SUC3* corpus with the *LäsBarT* corpus. To compare

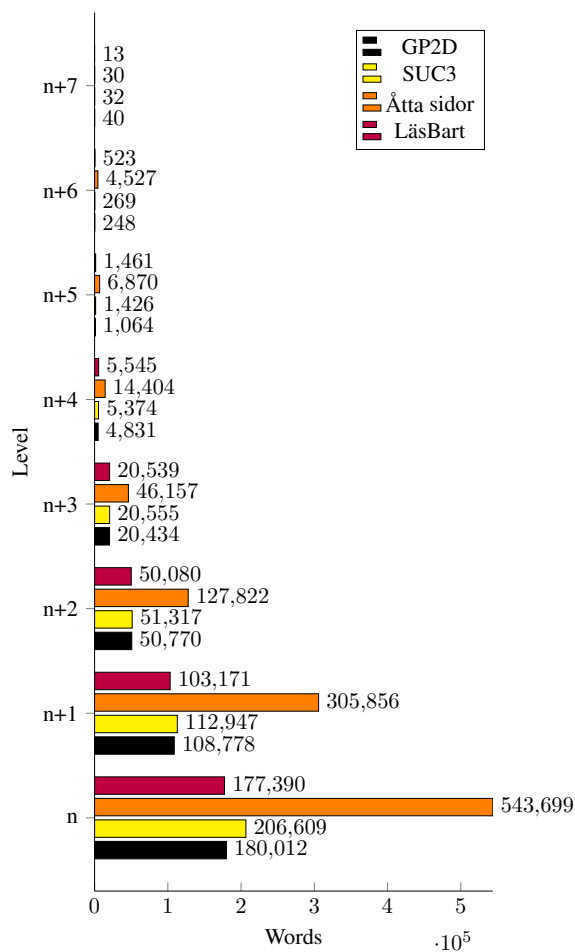


Figure 1: Number of words in the corpora at the various levels. Words at level n are the words in the corpora.

²<https://spraakbanken.gu.se/verktyg/korp/korpusstatistik>

the medians, a Mann-Whitney U test was performed. On average, the words of the *SUC3* corpus had a slightly lower number of steps to the top-level noun ($M = 0.93, Md = 1.0$) than the words of the *LäSBarT* corpus ($M = 1.02, Md = 1.0$). This difference was significant ($U = 17489728875.50, n1 = 206,609, n2 = 177,390, p < 0.001, cles = 0.32$).

In the second study, we compared corpora of the same genre (news texts): *GP2D* and *8 Sidor*. To compare the medians, a Mann-Whitney U test was performed. On average, the words of the *GP2D* corpus had a slightly higher number of steps to the top-level noun ($M = 1.03, Md = 1.0$) than the words of the *8 Sidor* corpus ($M = 0.93, Md = 1.0$). This difference was significant ($U = 46166030968.50, n1 = 180,012, n2 = 543,699, p < 0.001, cles = 0.37$).

The analyses of the relative frequencies of the corpora are presented in Table 1. The words at level n are the words that appear in the corpora³, and each $n+i$ step refers to the superordinate words. Three of the corpora (*LäSBarT*, *GP2D* and *8 Sidor*) had words represented at the level $n+8$, but since these words were very few (1, 4 and 1 words respectively), they were excluded from the analysis.

The *SUC3* corpus had the highest relative frequencies at level $n+3$. The *LäSBarT* corpus had the highest relative frequencies at level n . The *GP2D* corpus had the highest relative frequencies at level $n+7$. The *8 Sidor* corpus had the highest relative frequencies at level $n+3$.

All corpora, except for the *LäSBarT* corpus exhibited a tendency of peaking at level $n+3$ (see Table 1 and Figure 2).

Regarding the news corpora, we can see that the *8 Sidor* corpus has the highest relative frequency at level n , while the highest relative frequency at the standard news corpus *GP2D* is found at level $n+4$.

³We use the notation *level n* to describe the words of the corpora instead of, for example, level 0 words, as we do not know on what level of inclusiveness they actually appear. The words at level n are the words as they appear in the corpora, thus, they could be anywhere on the vertical axis of inclusiveness of the category. The only thing we know is the number of superordinate words, and therefore we chose to use the notation n for the corpus-level and $n+i$ for each superordinate level.

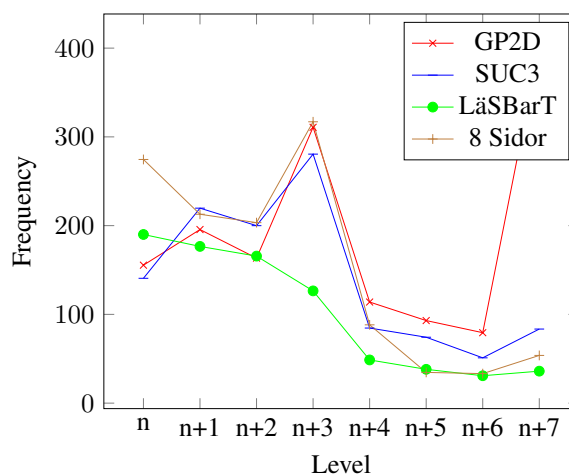


Figure 2: Relative frequencies at each level of the word hierarchy in the corpora.

3.3 Implications for Synonym Replacement Algorithms

From the research on cognitive linguistics referred above, we learnt that basic-level words are more frequently occurring in language, and often monolexemic. Thus, an algorithm shall reward synonym candidates that have high relative frequency and consist of one single lexeme; being monolexemic. To account for the monolexems, information from the frequency corpus about whether or not the word could be interpreted as a compound can be used.

From the corpus analysis, we also found that in the two standard corpora, there seems to be a frequency peak at level $n+3$. This could be due to the fact that when climbing higher up in the hierarchy of superordinate words, more general words are found, as these words are often more frequently occurring than words with a more specific meaning. When searching for synonyms, we hypothesise that the more general words are not necessarily good synonym candidates. For instance, whereas *horse* can be a good-enough synonym candidate for the word *shetland pony*, the word *animal* might be too general. We conducted experiments with varying levels and chose to restrict our synonym-seeking algorithm to not go beyond level $n+2$.

4 Synonym Replacement

Based on the analysis presented in Section 3.3, we developed an algorithm for choosing the best synonym from the extracted nouns and their superordinate words.

	SUC3	LäsBarT	GP2D	8 Sidor
Level n	140.66	190.02	155.44	274.54
Level n+1	219.69	176.59	195.59	212.82
Level n+2	199.97	165.67	163.39	203.38
Level n+3	280.56	126.48	310.78	317.01
Level n+4	84.60	48.68	113.92	88.22
Level n+5	74.25	38.10	93.04	34.64
Level n+6	51.04	30.88	79.37	33.24
Level n+7	83.47	36.03	401.41	53.76

Table 1: Average relative frequencies at each level of the words of the corpora. Highest level frequencies in boldface.

The resulting algorithm is presented in Algorithm 1. It picks, from words at most two levels up in the hierarchy, the most frequent monolexicemic word, if such exists, otherwise it picks the most frequent word.

Data: candidates: a word chain containing the word of the corpus and the superordinate words collected from Saldo.

Result: best synonym from candidates
 candidates.sort(key=frequency);
 bestSynonym = candidates[0];

```

for word in candidates[:3] do
  | if word is monolexicemic then
  | | bestSynonym = word;
  | | break;
  | end
end

```

Algorithm 1: The FM algorithm for choosing synonym.

5 Assessment of Synonym Replacement Algorithm

We compared the performance of our combined frequency/monolexemicity algorithm (hereafter: *FM*) with two baseline algorithms. The first baseline (*OneLevel*) always chose the word one level higher up in the hierarchy as the best synonym. If there was no superordinate word, the word remained unchanged. The second baseline (*Freq*) always chose the word with the overall highest relative frequency as the best synonym, thus disregarding the monolexemicity information.

We ran all algorithms on the nouns extracted from the standard corpora: *SUC3* and *GP2D*.

The results from both corpora regarding number of monolexicemic and polylexemic words are pre-

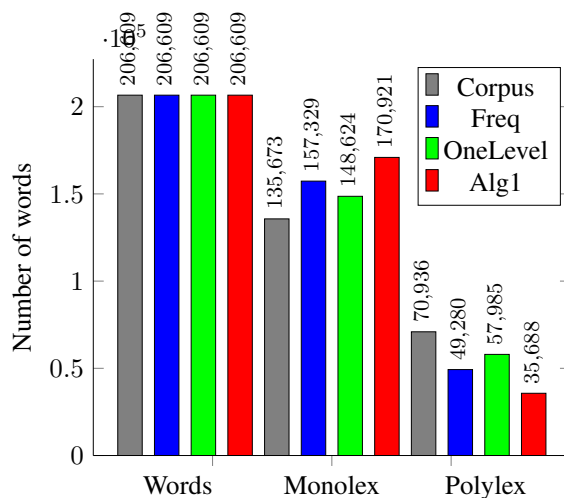


Figure 3: Number of total words, monolexicemic words, and polylexemic words in the SUC3 corpus after applying the algorithms. *Corpus* denotes the original values of the specific corpus.

sented in Figure 3 and Figure 4 respectively. The relative frequencies after running the algorithms are illustrated in Figure 5.

Regarding the *SUC3* corpus, all synonym replacement algorithms increased the number of monolexicemic words. The largest increase was observed for the FM algorithm (+35,248), followed by Freq (+21,656), and OneLevel (+12,951). Regarding the relative frequencies, all algorithms increased the average relative frequency of the exchanged words. The largest increase was seen for Freq (+153.68), followed by FM (+120.92), and OneLevel (+34.68).

On the *GP2D* corpus, the number of monolexicemic words increased for all algorithms. The largest increase was seen for the FM algorithm (+30,783), followed by the Freq algorithm (+21,091), and OneLevel (+9,482). All synonym

Example word chain	FM	OneLevel	Freq
procent - hundradel - bråkdel - del <i>percent - centesimal - fraction - part</i>	procent	hundradel	del
universitet - högskola - skola <i>university - college - school</i>	universitet	högskola	universitet
rubel - myntenhet - mynt - pengar <i>ruble - currency unit - coin - money</i>	mynt	myntenhet	mynt

Table 2: Example synonyms chosen by the different algorithms

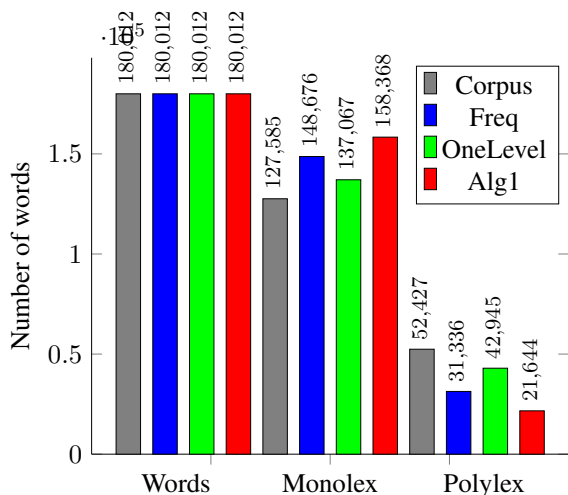


Figure 4: Number of total words, monolexic words, and polylexic words in the GP2D corpus after applying the algorithms. *Corpus* denotes the original values of the specific corpus.

replacement algorithms resulted in a higher average relative frequency, and the largest increase was observed for the Freq algorithm (+149.54), followed by the FM algorithm (+110.58), and OneLevel (+7.2).

Table 2 displays examples of the synonyms chosen by the respective algorithms. As can be seen frequency can sometimes choose a too general word, *del*, whereas OneLevel can pick a too specific word, *myntenhet*.

6 Discussion

The algorithm for finding synonyms proposed in this article is built on theory and corpus studies. This algorithm obviously needs to be evaluated and compared to other methods for extracting synonyms from corpora and lexical resources. It would be valuable to compare the algorithm with synonyms from, for example, the SynLex lexicon, and to evaluate whether the exchanged synonyms are simpler, when consulting lexicons of base vo-

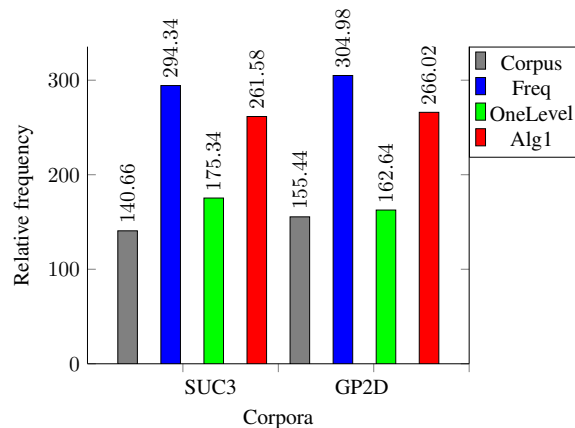


Figure 5: Relative frequencies for each corpus after applying the algorithms. *Corpus* denotes the original values of the specific corpus.

cabularies, as well as humans. It can also be enhanced with techniques to utilise semantic and synonym similarity (Kann and Rosell, 2005).

The corpus analyses were not conclusive, and, although further analyses will probably not present results that argues against the proposed algorithm, further investigations may be important for the study of language use and we therefore present a more detailed discussion on the corpus study.

We hypothesised that simple texts would exhibit a tendency towards the use of more basic-level words, when compared with texts written in standard Swedish. However, there was no clear support for this hypothesis. In the statistical analysis, we compared very large samples, and the presence of statistical significance is not surprising. When comparing the means and medians of the datasets, it is clear that the differences are small and the results should be interpreted with caution.

The results of the first study revealed that the *SUC3* corpus had a significantly lower average number of steps to the top-level noun, than the *LäsBarT* corpus. Since our hypothesis was that

the texts of the corpus of simple text would have a lower average number of steps to the top-level noun, these results showed a difference in the opposite direction.

The second study was normalised for genre, in the sense that the compared corpora contained texts of the same genre. The simple news corpus *8 Sidor* had a significantly lower number of steps to the top-level noun than the standard news corpus *GP2D*. This tendency is further supported by the results of the relative frequency analysis, where we clearly see that the *8 Sidor* corpus has relatively high average relative frequency at the base level (level n), although exhibiting the highest frequencies at level $n+3$, whereas the *GP2D* corpus generally had lower average frequencies at level n and the highest frequencies at level $n+7$.

Regarding the analyses of the relative frequencies, we would expect the standard corpora to have lower relative frequencies at the base level (level n) than the corpora of simple text. This difference can be observed in the *LäSBarT* corpus, which had the highest relative frequency scores at level n , but is less prominent in the *8 Sidor* corpus. However, even if the *8 Sidor* corpus exhibits the highest relative frequencies at level $n+3$, it is noteworthy that the frequencies are relatively high even at the lower levels. The level n score is the second highest frequency score for this corpus, and much higher when compared to the level n score of the standard corpus of the same genre, *GP2D*.

The *GP2D* corpus had the highest average frequency at level $n+7$, indicating that the words used in this corpus are more specific than in the other corpora. However, it should be noted that this high relative frequency score is based on a relatively low number of words (40), and that this corpus also exhibit the frequency peak at level $n+3$.

For *SUC3* and *8 Sidor*, the most frequent words are found at level $n+3$. This would mean that the more basic-level nouns could be found if we choose the superordinate words three levels above the original word. However, it could also indicate that the words at this level are higher up at Rosch's vertical axis, thus being more inclusive than the basic-level words, and therefore more frequent (compare: *shetland pony, horse, animal*).

When designing this study, we made a number of assumptions that can be discussed, such as the assumption of the nature of texts in simple Swedish versus texts in standard Swedish. We

made the assumption, according to Rosch's claims of basic-level terms, that the proportion of such constructions would be higher in the simple corpora. This assumption should be tested, for example by counting the relative frequencies of some base vocabulary list words (Heimann Mühlenbock and Johansson Kokkinakis, 2012) in both corpora.

The usage-based thesis of cognitive linguistics implies that we gain knowledge about the linguistic system by studying authentic language in use. To this background, it seems reasonable that a corpus study would be suitable for studying linguistic phenomena. However, there are some drawbacks of using such methods. One of the problems is that we worked with four very different corpora. Can we really say that a corpus reflects authentic and direct language use? For example, one commonly mentioned measure in this context is frequency. A frequency measure can provide information on how commonly used certain linguistic constructions are. However, what we see clearly in this study is that if we compare corpora of different characteristics, the frequency measures will differ between corpora depending on text type. A corpus of medical texts will have frequent constructions that do not even exist in a corpus of children's literature. The same issue will probably be manifested if we compare texts of different linguistic activities, such as spoken language with written language. This means that the insights that we can draw of the cognitive processes underlying the studied linguistic phenomenon will be very specific to the kind of corpus that we study. To compare corpora, we must make sure that the corpora are comparable, and consider the factor of language use reflected in the texts of the corpora when generalising our findings to a larger context.

7 Conclusion

The aim of this paper was to develop an algorithm for synonym replacement based on theories of basic-level nouns. We also presented results from a study exploring whether corpora of simple texts contain a higher proportion of basic-level nouns than corpora of standard Swedish, and to see how this could be used in the context of lexical text simplification.

We observed that the corpus of simple news text did indeed include more basic-level nouns than the corpus of standard news. This in turn shows that lexical simplification, through the use of base-

level nouns, may benefit from traversing a word hierarchy upwards. This could serve as a complement to the often-used replacement methods that rely on word length and word frequency measures.

We presented techniques for finding the best synonym candidate in a given word hierarchy, based on information about relative frequencies and monolexemy. We saw that all synonym replacement techniques, including the baseline methods, increased the number of monolexic words and relative frequencies. The FM algorithm aimed to reward high relative frequencies and monolexemy, while not climbing the word hierarchy too high, and seems to perform well with respect to these criteria. Future work includes further evaluation of this algorithm, and comparison to other synonym replacement strategies.

References

- David Alfter. 2021. *Exploring natural language processing for single-word and multi-word lexical complexity from a second language learner perspective*. Ph.D. thesis, Department of Swedish, University of Gothenburg, Gothenburg, Sweden.
- Sandra Aluisio, Lucia Specia, Caroline Gasperin, and Carolina Scarton. 2010. Readability assessment for text simplification. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9. Association for Computational Linguistics.
- Sandra M Aluísio, Lucia Specia, Thiago AS Pardo, Erick G Maziero, and Renata PM Fortes. 2008. Towards Brazilian Portuguese automatic text simplification systems. In *Proceedings of the eighth ACM symposium on Document engineering*, pages 240–248. ACM.
- Gianni Barlacchi and Sara Tonelli. 2013. Ernesta: A sentence simplification tool for children’s stories in Italian. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 476–487.
- Or Biran, Samuel Brody, and Noémie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 496–501.
- Lars Borin, Marcus Forsberg, and Lennart Lönngrén. 2008. SALDO 1.0 (Svenskt associationslexikon version 2). *Språkbanken, Göteborgs universitet*.
- Lars Borin and Markus Forsberg. 2014. Swesaurus; or, The Frankenstein approach to Wordnet construction. In *Proceedings of the Seventh Global Wordnet Conference*, pages 215–223.
- Arnaldo Candido Jr, Erick Maziero, Caroline Gasperin, Thiago AS Pardo, Lucia Specia, and Sandra M Aluisio. 2009. Supporting the adaptation of texts for poor literacy readers: a text simplification editor for Brazilian Portuguese. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 34–42. Association for Computational Linguistics.
- Yvonne Canning and John Tait. 1999. Syntactic simplification of newspaper text for aphasic readers. In *ACM SIGIR’99 Workshop on Customised Information Delivery*.
- John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of English newspaper text to assist aphasic readers. In *Proceedings of the AAAI98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, volume 1, pages 7–10. Citeseer.
- Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and Methods for Text Simplification. In *Proceedings of the Sixteenth International Conference on Computational Linguistics (COLING ’96)*.
- Jin-Woo Chung, Hye-Jin Min, Joonyeob Kim, and Jong C Park. 2013. Enhancing readability of web documents by text augmentation for deaf people. In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*, pages 1–10.
- Walter Daelemans, Anja Höthker, and Erik F Tjong Kim Sang. 2004. Automatic sentence simplification for subtitling in Dutch and English. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*.
- Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Proceedings of the SIGIR workshop on accessible search systems*, pages 19–26. ACM; New York.
- Siobhan Devlin and Gary Unthank. 2006. Helping aphasic people process online information. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, pages 225–226.
- Biljana Drndarević and Horacio Saggon. 2012. Towards automatic lexical simplification in Spanish: an empirical study. In *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*, pages 8–16.
- Eva Ejerhed, Gunnel Källgren, and Benny Brodda. 2006. Stockholm Umeå Corpus version 2.0.
- Vyvyan Evans. 2019. *Cognitive Linguistics (2nd edition)*. Edinburgh: Edinburgh University Press.
- Katarina Heimann Mühlenbock and Sofie Johansson Kokkinakis. 2012. SweVoc - a Swedish vocabulary resource for CALL. In *Proceedings of the*

SLTC 2012 workshop on NLP for CALL, pages 28–34, Lund. Linköping University Electronic Press.

Firas Hmida, Mokhtar B. Billami, Thomas François, and Núria Gala. 2018. Assisted lexical simplification for French native children with reading difficulties. In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, pages 21–28, Tilburg, the Netherlands. Association for Computational Linguistics.

Kentaro Inui, Atsushi Fujita, Tetsuro Takahashi, Ryū Iida, and Tomoya Iwakura. 2003. Text simplification for reading assistance: a project note. In *Proceedings of the second international workshop on Paraphrasing*, pages 9–16. Association for Computational Linguistics.

Viggo Kann and Magnus Rosell. 2005. Free construction of a free Swedish dictionary of synonyms. In *Proceedings of the 15th NODALIDA conference*, pages 105–110, Stockholm.

MTM. 2020. Att skriva lättläst. <https://www.mtm.se/var-verksamhet/lattlast/att-skriva-lattlast/>. Accessed: 2020-10-05.

Katarina Mühlenbock. 2008. Readable, Legible or Plain Words – Presentation of an easy-to-read Swedish corpus. In *Multilingualism: Proceedings of the 23rd Scandinavian Conference of Linguistics*, volume 8 of *Acta Universitatis Upsaliensis*, pages 327–329, Uppsala, Sweden. Acta Universitatis Upsaliensis.

Gustavo Henrique Paetzold. 2016. *Lexical Simplification for Non-Native English Speakers*. Ph.d. thesis, University of Sheffield, Sheffield, UK.

Sarah E Petersen and Mari Ostendorf. 2007. Text simplification for language learners: a corpus analysis. In *Workshop on Speech and Language Technology in Education*.

Eleanor Rosch, Carolyn B. Mervis, Wayne D. Gray, David M. Johnson, and Penny Boyes-braem. 1976. Basic objects in natural categories. *Cognitive Psychology*.

SuperSim: a test set for word similarity and relatedness in Swedish

Simon Hengchen, Nina Tahmasebi
Språkbanken Text, Department of Swedish
University of Gothenburg

{simon.hengchen;nina.tahmasebi}@gu.se

Abstract

Language models are notoriously difficult to evaluate. We release SuperSim, a large-scale similarity and relatedness test set for Swedish built with expert human judgments. The test set is composed of 1,360 word-pairs independently judged for both relatedness and similarity by five annotators. We evaluate three different models (Word2Vec, fastText, and GloVe) trained on two separate Swedish datasets, namely the Swedish Gigaword corpus and a Swedish Wikipedia dump, to provide a baseline for future comparison. We release the fully annotated test set, code, baseline models, and data.¹

1 Introduction

It is said that a *cup* and *coffee* are not very similar while *car* and *train* are much more so given that they share multiple similar features. Instead, *cup* and *coffee* are highly related, as we typically enjoy the one in the other. Of course, an immediate question that arises is whether we have words that are similar but not related? Existing similarity datasets have tended to rate words for their similarity, relatedness, or a mixture of both, but not either or. However, without both kind of information, we cannot know if words are related but not similar, or similar but not related.

The most common motivation for using word similarity datasets, such as SimLex-999 (Hill et al., 2015) and WordSim353 (Finkelstein et al., 2001), is for use as a quality check for word embedding models. The aim of most embedding models is to capture a word’s semantic relationships, such that words that are similar in meaning are placed close in the semantic space; foods

with other foods, technical terms together and separated from the musical instruments, to give an example. However, the optimal performance of such a semantic space is judged by whether or not one wishes to capture similarity of words, or relatedness. It seems obvious that presenting *cup* as a query reformulation for *coffee* in information retrieval seems off, while presenting *lamborghini* when searching for *ferrari* can be completely acceptable. Inversely, in places where relatedness is needed, offering a *cup* when one asks for a *coffee* is correct.

While the first word similarity datasets appeared for English, in the past few years we have seen datasets for a range of different languages (see Section 2). For Swedish, there exists one automatically-created resource based on an association lexicon by Fallgren et al. (2016). However, there are to date no test sets that are (1) expertly-annotated, (2) comparable to other international test sets, and (3) annotated for both relatedness and similarity. And because we cannot know which motivation lies behind creating a vector space, and because both relatedness and similarity seem equally valid, we have opted to create *SuperSim*. The SuperSim test set is a larger-scale similarity and relatedness set for Swedish, consisting of 1,301 words and 1,360 pairs rated by 5 expert annotators. The pairs are based on SimLex-999 and WordSim353, and can be used to assess the performance of word embedding models, but also answer questions as to whether words are likely to be similar but not related.

2 Related Work

Several works aim to provide test sets to assess the quality of word embedding models. Most of them tackle English (Rubenstein and Goodenough, 1965; Miller and Charles, 1991; Agirre et al., 2009; Bruni et al., 2012; Hill et al., 2015). Russian, Italian and German are cov-

¹<https://zenodo.org/record/4660084>.

ered by Leviant and Reichart (2015) who translated the pairs in WordSim353 and SimLex-999, and asked crowdworkers to judge them on a 0-10 scale. The SemEval-2017 Task 2 on Multilingual and Cross-lingual Semantic Word Similarity (Camacho-Collados et al., 2017) provides pairs in 5 languages: English, Farsi, German, Italian and Spanish. Ercan and Yıldız (2018) provide 500 word pairs in Turkish annotated by 12 humans for both similarity and relatedness on a scale ranging from 0 to 10, while Finnish is covered in Venekoski and Vankka (2017). More recently, Multi-SimLex (Vulić et al., 2020) provides annotations in Mandarin Chinese, Yue Chinese, Welsh, English, Estonian, Finnish, French, Hebrew, Polish, Russian, Spanish, Kiswahili, and Arabic, with open guidelines and encouragement to join in with more languages.²

For Swedish, Fallgren et al. (2016) harness the Swedish Association Lexicon SALDO (Borin et al., 2013), a large lexical-semantic resource that differs much from Wordnet (Fellbaum, 1998) insofar as it organises words mainly with the ‘association’ relation. The authors use SALDO’s ‘super-senses’ to adapt Tsvetkov et al. (2016)’s QVEC-CCA intrinsic evaluation measure to Swedish. Still on evaluating Swedish language models, Adewumi et al. (2020b) propose an analogy test set built on the one proposed by Mikolov et al. (2013), and evaluate common architectures on downstream tasks. The same authors further compare these architectures on models trained on different datasets (namely the Swedish Gigaword corpus (Rødven-Eide et al., 2016) and the Swedish Wikipedia) by focusing on Swedish and utilising their analogy test set (Adewumi et al., 2020a). Finally, for Swedish, SwedishGLUE/SuperLim³ (Adesam et al., 2020) is currently being developed as a benchmark suite for language models in Swedish, somewhat mirroring English counterparts (Wang et al., 2018, 2019).

Whether similarity test sets actually allow to capture and evaluate lexical semantics is debatable (Faruqui et al., 2016; Schnabel et al., 2015). Nonetheless, they have the advantage of providing a straightforward way of optimising word embeddings (through hyper-parameter search, at

²The website is updated with new annotations: <https://multisimlex.com/>.

³<https://spraakbanken.gu.se/projekt/superlim-en-svensk-testmangd-for-sprakmodeller>

the risk of overfitting), or to be used more creatively in other tasks (Dubossarsky et al., 2019) where “quantifiable synonymy” is required. Finally, task-specific evaluation (as recommended by (Faruqui et al., 2016)) is, for languages other than English, more than often nonexistent – making test sets such as the one presented in this work a good alternative.

Our dataset differs from previous work in the sense that it provides expert judgments for Swedish for both relatedness and similarity, and hence comprises two separate sets of judgments, as done by skilled annotators.⁴ A description of the procedure is available in Section 3.

2.1 Relatedness and Similarity

Our work heavily draws from Hill et al. (2015), who made a large distinction between relatedness and similarity. Indeed, the authors report that previous work such as Agirre et al. (2009) or Bruni et al. (2012) do not consider relatedness and similarity to be different. Words like *coffee* and *cup*, to reuse the example by Hill et al. (2015), are obviously related (one is used to drink the other, they can both be found in a kitchen, etc.) but at the same time dissimilar (one is (...usually) a liquid and the other is a solid, one is ingested and not the other, etc.).

All pairs in SuperSim are independently judged for similarity and relatedness. To explain the concept of similarity to annotators, we have reused the approach of Hill et al. (2015) who introduced it via the idea of synonymy, and in contrast to association: “In contrast, although the following word pairs are related, they are not very similar. The words represent entirely different types of things.” They further give the example of “car / tyre.” We use this definition embedded in the SimLex-999 guidelines to define relatedness according to the following: “In Task 2, we also ask that you rate the same word pairs for their relatedness. For this task, consider the inverse of similarity: *car* and *tyre* are related even if they are not synonyms. However, synonyms are also related.”

⁴We have opted not to follow Multi-SimLex because (1) we want to have annotations for both relatedness and similarity, and (2) we have limited possibility to use platforms such as Amazon Mechanical Turk, and have thus resorted to using skilled annotators: to illustrate, we are bound to the hourly rate of 326 SEK (32.08 EUR). As a result the cost of annotating with 10 annotators is significantly higher, in particular if we want two separate sets of annotations.

3 Dataset description

While the WordSim353 pairs were chosen for use in information retrieval and to some extent mix similarity and relatedness, the original SimLex-999 pairs were chosen with more care. They were meant to measure the ability of different models to capture similarity as opposed to association, contain words from different part-of-speech (nouns, verbs, and adjectives), and represent different concreteness levels. Despite the risks of losing some intended effect in translation, we opted to base SuperSim on both of these resources rather than start from scratch.

3.1 Methodology

We machine-translated all words in WordSim353 and SimLex-999 to Swedish. The translations were manually checked by a semanticist who is a native speaker of Swedish, holds an MA in linguistics, and is currently working towards obtaining a PhD in linguistics. The semanticist was presented a list of words, out of context, decoupled from the pairs they were parts of. Where needed, translations were corrected. Pairs were reconstructed according to the original datasets, except for the few cases where the translation process would create duplicates. In a few cases where one single translation was not obvious – i.e. cases where either Google Translate or the semanticist would output two (equally likely) possible Swedish translations for the same English word –, two pairs were constructed: one with each possible translation. For example, the presence of ‘drug’ led to pairs with both the *läkemedel* (a medical drug aimed at treating pathologies) and *drog* (a narcotic or stimulant substance, usually illicit) translations.

We selected 5 annotators (4F/1M) who are native speakers of Swedish and all have experience working with annotation tasks. One of the annotators was the same person who manually checked the correctness of the translations. The other 4 annotators can be described as follows:

- holds an MA in linguistics and has experience in lexicography,
- holds an MA in linguistics,
- holds BAs in linguistics and Spanish and is studying for an MSc in language technology,
- holds a BA in linguistics and has extensive work experience with different language-

related tasks such as translation and NLP (on top of annotation).

Annotators were each given (i) the original SimLex-999 annotation instructions containing examples illustrating the difference between relatedness and similarity; (ii) one file for the relatedness scores; and (iii) one file for the similarity scores. They were instructed to complete the annotation for similarity before moving on to relatedness, and complied. The annotation took place, and was monitored, on Google Sheets. Annotators did not have access to each others’ sheets, nor were they aware of who the other annotators were.

To allow for a finer granularity as well as to echo previous work, annotators were tasked with assigning scores on a 0-10 scale, rather than 1-6 as in SimLex-999. Unlike the procedure for Simlex, where sliders were given (and hence the annotators could choose real values), our annotators assigned discrete values between 0–10. This procedure resulted in pairs with the same score, and thus many rank ties.

3.2 SuperSim stats

The entire SuperSim consists of 1,360 pairs. Out of these, 351 pairs stem from WordSim353 and 997 pairs from SimLex-999. Pairs where both words translate into one in Swedish are removed from the SimLex-999 and WordSim353 subsets, thus resulting in fewer pairs than the original datasets: for example, ‘engine’ and ‘motor’ are both translated as *motor* and therefore the ‘motor’ – ‘engine’ pair is removed. The SuperSim set consists of both sets, as well as of a set of additional pairs where multiple translations were used (see the *läkemedel* and *drog* example above). The full set of 1,360 pairs is annotated for both similarity and relatedness separately, resulting in a total of $2 * 1,360$ gold scores, and thus 13,600 individual judgments. An example of relatedness judgments for two pairs is available in table form in Table 1.

We release two tab-separated files (one for relatedness, one for similarity) containing judgments from all annotators as well as the mean gold score. We additionally release all baseline models, code, and pre-processed data where permissible. The data is freely available for download at <https://zenodo.org/record/4660084>.

Table 1: Example of relatedness judgments on pairs *flicka-barn* ‘girl-child’ and *skola-mitten* ‘school-centre.’

Word 1	Word 2	Anno 1	Anno 2	Anno 3	Anno 4	Anno 5	Average
flicka	barn	10	10	10	8	10	9.6
skola	mitten	1	0	0	0	0	0.2

3.3 Intra-rater agreement

For quality control, annotation files contained a total of 69 randomly sampled duplicate pairs, in addition to the 1,360 true pairs.⁵ These duplicates allowed us to calculate every annotator’s consistency, and to judge how difficult each task was in practice. Table 2 illustrates the consistency of every annotator in the similarity and relatedness tasks for our 69 control pairs. ‘Disagreement’ indicates two different values for any given pair and ‘hard disagreement’ two values with an absolute difference higher than 2 (on the scale of 0–10). On average, the hard disagreements differed by 4.3 points for relatedness, and by 3.0 for similarity, and there were more disagreements (both kinds) for relatedness, indicating that for humans, relatedness is the harder task. In addition, we indicate the computed self-agreement score (Krippendorff’s alpha, Krippendorff 2018) for every annotator for both tasks. Despite annotators disagreeing somewhat with themselves, Krippendorff’s alpha indicates they annotated word pairs consistently.

Out of the 69 control pairs, 4 were inconsistently annotated by four annotators for similarity, while 12 pairs were inconsistently annotated by four or more annotators for relatedness: 3 by all five annotators, and 9 by four. The three “hardest” pairs to annotate for relatedness are *lycklig-arg* ‘happy-angry,’ *sommar-natur* ‘summer-nature,’ *tillkännagivande-varning* ‘announcement-warning.’

3.4 Inter-rater agreement

Following Hill et al. (2015), we use the average Spearman’s ρ for measuring inter-rater agreement by taking the average of pairwise Spearman’s ρ correlations between the ratings of all respondents.⁶ For the original SimLex-999, over-

⁵SuperSim includes the values for the first seen annotation of a duplicate pair. To illustrate: if a control pair was annotated first to have a score of 3 and then to have a score of 6, the first score of 3 is kept.

⁶We use the `scipy.stats.mstats.spearmanr` (Virtanen et al., 2020) implementation with rank ties.

all agreement was $\rho = 0.67$ as compared to WordSim353 where $\rho = 0.61$ using the same method. Spearman’s ρ for our similarity rankings is 0.67. In addition, we have a Spearman’s ρ for our relatedness rankings of 0.73.⁷ It is unclear how the background of our annotators affects the quality of their annotation. In another semantic annotation study, although on historical data, Schlechtweg et al. (2018) show a larger agreement between annotators sharing a background in historical linguistics than between a historical linguist and a ‘non-expert’ native speaker. It is, however, fully possible that the linguistic expertise of the annotators affects the similarity and relatedness judgments in a negative way. We leave this investigation for further work.

4 Model evaluation

To provide a baseline for evaluation of embedding models on SuperSim, we trained three different models on two separate datasets.

4.1 Baseline Models

We chose three standard models, Word2Vec (Mikolov et al., 2013), fastText (Bojanowski et al., 2017), and GloVe (Pennington et al., 2014). Word2Vec and fastText models are trained with gensim (Řehůřek and Sojka, 2010) while the GloVe embeddings are trained using the official C implementation provided by Pennington et al. (2014).⁸

4.2 Training data

We use two datasets. The largest of the two comprises the Swedish Culturomics Gigaword corpus (Rødven-Eide et al., 2016), which con-

⁷These results are opposing those of the disagreements which indicate that similarity is easier than relatedness for our annotators. We postulate that this can be due to the many rank ties we have in the similarity testset (where many pairs have 0 similarity). If we use the Pearson’s ρ , we get values of $\rho = 0.722$ for relatedness, and $\rho = 0.715$ for similarity bringing the two tasks much closer.

⁸Tests were also made using the Python implementation available at <https://github.com/maciejkula/glove-python>, with similar performance.

Table 2: Number of control word-pairs with annotator self-disagreements. ‘Disagreement.’ = different values between two annotations for a given pair (0-10 scale), ‘hard disagreement.’ = difference > 2 between values between two annotations for a given pair (0-10 scale), α = Krippendorff’s alpha. Total number of control pairs is 69, percentages follow absolute counts in parentheses.

	Consistency of judgments					
	Similarity			Relatedness		
	# disagree. (%)	# hard disagree. (%)	α	# disagree. (%)	# hard disagree. (%)	α
Anno 1	17 (25%)	5 (7%)	0.83	20 (29%)	10 (14%)	0.89
Anno 2	1 (1%)	1 (1%)	0.99	26 (38%)	11 (16%)	0.86
Anno 3	21 (30%)	6 (9%)	0.94	24 (35%)	9 (13%)	0.87
Anno 4	10 (14%)	0 (0%)	0.96	18 (26%)	4 (8%)	0.96
Anno 5	29 (42%)	3 (4%)	0.89	28 (41%)	7 (10%)	0.89

Table 3: Evaluation of models trained on the Swedish Gigaword corpus. WordSim353 and SimLex-999 are subsets of the SuperSim. Best results for each “test set - task” combination are bolded.

Model	Test set	Spearman’s ρ relatedness	Spearman’s ρ similarity	Included pairs
Word2Vec	SuperSim	0.539	0.496	1,255
	WordSim353 pairs	0.560	0.453	325
	SimLex-999 pairs	0.499	0.436	923
fastText	SuperSim	0.550	0.528	1,297
	WordSim353 pairs	0.547	0.477	347
	SimLex-999 pairs	0.520	0.471	942
GloVe	SuperSim	0.548	0.499	1,255
	WordSim353 pairs	0.546	0.435	325
	SimLex-999 pairs	0.516	0.448	923

tains a billion words⁹ in Swedish from different sources including fiction, government, news, science, and social media. The second dataset is a recent Swedish Wikipedia dump with a total of 696,500,782 tokens.¹⁰

While the Swedish Gigaword corpus contains text from the Swedish Wikipedia, Rødven-Eide et al. (2016) precise that about 150M tokens out of the 1G in Gigaword (14.9%) stem from the Swedish Wikipedia. In that respect, there is an overlap in terms of content in our baseline corpora. However, as the Swedish Wikipedia has grown extensively over the years and only a sub-part of it was used in in Rødven-Eide et al. (2016), the overlap is small and we thus have opted to also use the Gigaword corpus as it is substantially larger and contains other genres of text.

The Wikipedia dump was processed with a version of the Perl script released by Matt Mahoney¹¹

⁹1,015,635,151 tokens in 59,736,642 sentences, to be precise.

¹⁰Available at <https://dumps.wikimedia.org/svwiki/20201020/svwiki-20201020-pages-articles.xml.bz2>.

¹¹The script is available at <http://mattmahoney.net/dc/textdata.html>. It effectively only keeps what should be displayed in a web browser

modified to account for specific non-ASCII characters (äåöé) and to transform digits to their Swedish written form (eg: 2 \rightarrow två).¹²

All baseline models are trained on lowercased tokens with default hyperparameters.¹³

4.3 Results

An overview of the performance of the three baseline models is available in Table 3 and Table 4. In both tables we show model performance on similarity and relatedness judgments. We split the results into three sets, one for the entire SuperSim, and two for its subsets: WordSim353 and SimLex-999. For each model and dataset, we present Spearman’s rank correlation ρ between the ranking produced by the model compared to the gold ranking in each testset (relatedness and similarity). As fastText uses subword information to build vectors, it deals better with out-of-vocabulary words, hence the higher number of

and removes tables but keeps image captions, while links are converted to normal text. Characters are lowercased.

¹²‘1’, which can be either *en* or *ett* in Swedish, was replaced by ‘ett’ every time.

¹³Except for $sg = 1$, $min_count = 100$ and $seed = 1830$.

Table 4: Evaluation of models trained on the Swedish Wikipedia. WordSim353 and SimLex-999 are subsets of the SuperSim. Best results for each “test set - task” combination are bolded.

Model	Test set	Spearman’s ρ relatedness	Spearman’s ρ similarity	Included pairs
Word2Vec	SuperSim	0.410	0.410	1,197
	WordSim353 pairs	0.469	0.415	315
	SimLex-999 pairs	0.352	0.337	876
fastText	SuperSim	0.349	0.365	1,297
	WordSim353 pairs	0.339	0.334	347
	SimLex-999 pairs	0.322	0.311	942
GloVe	SuperSim	0.467	0.440	1,197
	WordSim353 pairs	0.524	0.429	315
	SimLex-999 pairs	0.418	0.375	876

pairs included in the evaluation.

To provide a partial reference point, Hill et al. (2015) report, for Word2Vec trained on English Wikipedia, ρ scores of 0.655 on WordSim353, and 0.414 on SimLex-999.

From the results in Table 3 and 4, it appears that fastText is the most impacted by the size of the training data, as its performance when trained on the smaller Wikipedia corpus is ‘much’ lower than on the larger Gigaword: 0.349 vs 0.550 for SuperSim relatedness and 0.365 vs 0.528 for SuperSim similarity – both tasks where fastText actually performs best on Gigawords out of the three models tested. We find that all models perform better when trained on Gigaword as compared to Wikipedia. Contrary to results on the analogy task reported by Adewumi et al. (2020a), our experiments on SuperSim seem to confirm the usual trope that training on more data indeed leads to **overall** better embeddings, as the higher scores, in terms of absolute numbers, are all from models trained on the larger Gigaword corpus. Nonetheless, the discrepancy between our results and theirs might be due to a range of factors, including pre-processing and hyperparameter tuning (which we did not do).¹⁴

Note that for similarity, Word2Vec trained on Gigaword performs slightly better on the translated SimLex-999 pairs (0.436) than Word2Vec does on English SimLex-999 (0.414) but substantially lower for WordSim (0.436 vs 0.655) (Hill et al., 2015). We make the comparison for Gigaword, rather than Wikipedia because of the com-

parable size, rather than the genre. This effect could be due to different pre-processing and model parameters used, but it could also be an effect of the multiple ties present in our test set. We do, however, consistently confirm the original conclusion: **SimLex-999 seems harder for the models than WordSim353.**

GloVe is the clear winner on the smaller Wikipedia dataset, where it outperforms the other two models for all test sets, and is on par with Word2Vec for Gigaword.

Overall, our results indicate that **for the tested models relatedness is an easier task than similarity**: every model – aside from fastText on SuperSim – performs better (or equally well) on relatedness on the whole test set, as well as on its subparts, compared to similarity.

5 Conclusions and future work

In this paper, we presented SuperSim, a Swedish similarity and relatedness test set made of new judgments of the translated pairs of both SimLex-999 and WordSim353. All pairs have been rated by five expert annotators, independently for both similarity and relatedness. Our inter-annotator agreements mimic those of the original test sets, but also indicate that similarity is an easier task to rate than relatedness, while our intra-rater agreements on 69 control pairs indicate that the annotation is reasonably consistent.

To provide a baseline for model performance, we trained three different models, namely Word2Vec, fastText and GloVe, on two separate Swedish datasets. The first comprises a general purpose dataset, namely the The Swedish Culturalomics Gigaword Corpus with different genres of text spanning 1950-2015. The second comprises

¹⁴The effect of the benefits of more training data is confounded with the broader genre definitions in Gigaword that could be an indication of the advantage of including e.g., fiction and social media text in defining for example emotions. We leave a detailed investigation into this for future work.

a recent Swedish Wikipedia dump. On the Gigaword corpus, we find that fastText is best at capturing both relatedness and similarity while for Wikipedia, GloVe performs the best.

Finally, to answer the question posed in the introduction: it is common to have words that are highly related, but not similar. To give a few examples, these are pairs with relatedness 10 and similarity 0: *bil-motorväg* ‘car-highway,’ *datum-kalender* ‘date-calendar,’ *ord-ordbok* ‘word-dictionary,’ *skola-betyg* ‘school-grade,’ and *tennis-racket* ‘tennis-racket.’

The opposite however, does not hold. Only four pairs have a similarity score higher than the relatedness score, and in all cases the difference is smaller than 0.6: *bli-verka* ‘become-seem,’ *rör-cigarr* ‘pipe-cigarr,’ *ståltråd-sladd* ‘wire-cord,’ *tillägna sig-skaffa sig* ‘get-acquire.’

For future work, the SuperSim testset can be improved both in terms of added annotations (more annotators), and with respect to more fine-grained judgements (real values in contrast to discrete ones currently used) to reduce the number of rank ties.

6 Acknowledgments

We would like to thank Tosin P. Adewumi, Lidia Pivovarova, Elaine Zosa, Sasha (Aleksandrs) Berdicevskis, Lars Borin, Erika Wauthia, Haim Dubossarsky, Stian Rødven-Eide as well as the anonymous reviewers for their insightful comments. This work has been funded in part by the project *Towards Computational Lexical Semantic Change Detection* supported by the Swedish Research Council (2019–2022; dnr 2018-01184), and *Nationella Språkbanken* (the Swedish National Language Bank), jointly funded by the Swedish Research Council (2018–2024; dnr 2017-00626) and its ten partner institutions.

References

Yvonne Adesam, Aleksandrs Berdicevskis, and Felix Morger. 2020. Swedishglue – towards a swedish test set for evaluating natural language understanding models. Technical report, University of Gothenburg.

Tosin P Adewumi, Foteini Liwicki, and Marcus Liwicki. 2020a. Corpora compared: The case of the swedish gigaword & wikipedia corpora. *arXiv preprint arXiv:2011.03281*.

Tosin P Adewumi, Foteini Liwicki, and Marcus Liwicki. 2020b. Exploring Swedish & English fasttext embeddings with the transformer. *arXiv preprint arXiv:2007.16007*.

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalová, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of NAACL-HLT*, pages 19–27.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Lars Borin, Markus Forsberg, and Lennart Lönngrén. 2013. Saldo: a touch of yin to wordnet’s yang. *Language resources and evaluation*, 47(4):1191–1211.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 136–145, Jeju Island, Korea. Association for Computational Linguistics.

Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. 2017. SemEval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 15–26, Vancouver, Canada. Association for Computational Linguistics.

Haim Dubossarsky, Simon Hengchen, Nina Tahmasebi, and Dominik Schlechtweg. 2019. Time-out: Temporal referencing for robust modeling of lexical semantic change. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 457–470, Florence, Italy. Association for Computational Linguistics.

Gökhan Ercan and Olcay Taner Yıldız. 2018. AnlamVer: Semantic model evaluation dataset for Turkish - word similarity and relatedness. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3819–3836, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Per Fallgren, Jesper Segeblad, and Marco Kuhlmann. 2016. Towards a standard dataset of Swedish word vectors. In *Sixth Swedish Language Technology Conference (SLTC), Umeå 17-18 nov 2016*.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35, Berlin, Germany. Association for Computational Linguistics.

Christiane Fellbaum. 1998. WordNet: An electronic lexical database. Christiane Fellbaum (Ed.). Cambridge, MA: MIT Press, 1998. Pp. 423. *Applied Psycholinguistics*, 22(01):131–134.

- Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414.
- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.
- Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.
- Ira Leviant and Roi Reichart. 2015. Separated by an un-common language: Towards judgment language informed vector space modeling. *arXiv preprint arXiv:1508.00106*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.
- Stian Rødven-Eide, Nina Tahmasebi, and Lars Borin. 2016. The Swedish culturomics gigaword corpus: A one billion word Swedish reference dataset for NLP. In *Digital Humanities 2016.*, 126, pages 8–12. Linköping University Electronic Press.
- Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.
- Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. 2018. Diachronic usage relatedness (DURel): A framework for the annotation of lexical semantic change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 169–174, New Orleans, Louisiana. Association for Computational Linguistics.
- Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon, Portugal. Association for Computational Linguistics.
- Yulia Tsvetkov, Manaal Faruqui, and Chris Dyer. 2016. Correlation-based intrinsic evaluation of word vector representations. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 111–115, Berlin, Germany. Association for Computational Linguistics.
- Viljami Venekoski and Jouko Vankka. 2017. Finnish resources for evaluating language model semantics. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 231–236, Gothenburg, Sweden. Association for Computational Linguistics.
- Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.
- Ivan Vulić, Simon Baker, Edoardo Maria Ponti, Ulla Petti, Ira Leviant, Kelly Wing, Olga Majewska, Eden Bar, Matt Malone, Thierry Poibeau, Roi Reichart, and Anna Korhonen. 2020. Multi-simlex: A large-scale evaluation of multilingual and cross-lingual lexical semantic similarity. *Computational Linguistics*, 0(0):1–51.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Super-glue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

NLI Data Sanity Check: Assessing the Effect of Data Corruption on Model Performance

Aarne Talman^{*†}, Marianna Apidianaki^{*}, Stergios Chatzikyriakidis[‡], Jörg Tiedemann^{*}

^{*}Department of Digital Humanities, University of Helsinki
{name.surname}@helsinki.fi

[†]Basement AI

[‡]CLASP, Department of Philosophy, Linguistics and Theory of Science, University of Gothenburg
{name.surname}@gu.se

Abstract

Pre-trained neural language models give high performance on natural language inference (NLI) tasks. But whether they actually understand the meaning of the processed sequences remains unclear. We propose a new diagnostics test suite which allows to assess whether a dataset constitutes a good testbed for evaluating the models’ meaning understanding capabilities. We specifically apply controlled corruption transformations to widely used benchmarks (MNLI and ANLI), which involve removing entire word classes and often lead to non-sensical sentence pairs. If model accuracy on the corrupted data remains high, then the dataset is likely to contain statistical biases and artefacts that guide prediction. Inversely, a large decrease in model accuracy indicates that the original dataset provides a proper challenge to the models’ reasoning capabilities. Hence, our proposed controls can serve as a crash test for developing high quality data for NLI tasks.

1 Introduction

Assessing the natural language inference (NLI) and understanding (NLU) capabilities of a model poses numerous challenges, one of which is the quality and composition of the data used for evaluation. Popular NLI datasets (Bowman et al., 2015; Marelli et al., 2014) contain annotation artefacts and statistical irregularities that can be easily grasped by a model during training and guide prediction, even if the model has not acquired the knowledge needed to perform this kind of reasoning. Notably, recent work shows that major modifications such as word shuffling do not hurt BERT’s (Devlin et al., 2019) NLU capabilities

	Premise	Hypothesis
Contradiction	He was hardly more than five feet , four inches , but carried himself with great dignity .	The man was 6 foot tall .
Entailment	Two plants died on the long journey and the third one found its way to Jamaica exactly how is still shrouded in mystery .	The third plant was a different type from the first two.
Neutral	In a couple of days the wagon train would head on north to Tucson , but now the activity in the plaza was a mixture of market day and fiesta .	They were south of Tucson .

Table 1: Sentence pairs from a corrupted MNLI training dataset where nouns have been removed.

much, mainly due to individual words’ impact on prediction (Pham et al., 2020). To the contrary, small tweaks or perturbations in the data, such as replacing words with mutually exclusive co-hyponyms and antonyms (Glockner et al., 2018) or changing the order of the two sentences (Wang et al., 2019b), has been shown to hurt the performance of NLI models.

Motivated by this situation, our goal is to contribute a new suite of diagnostic tests that can be used to assess the quality of an NLU benchmark. In particular, we conduct a series of controlled experiments where a set of data corruption transformations are applied to the widely used MNLI (Williams et al., 2018) and ANLI (Nie et al., 2020) datasets, and explore their impact on fine-tuned BERT and ROBERTa (Liu et al., 2019) model performance. The obtained results provide evidence that can reveal the quality of a dataset: Given that the transformations seriously affect the quality of NLI sentences, going as far as making them unintelligible (cf. examples in Table 1), a decrease in performance for models fine-tuned on the cor-

rupted dataset would be expected. High performance would, instead, indicate the presence of biases and other artefacts in the dataset which guide models' predictions. This situation would be indicative of a low quality dataset, i.e. one we cannot rely upon to draw safe conclusions about a model's NLI capabilities.

Bringing in additional evidence to the debate on problematic NLI evaluation setups and how poorly they represent the real inference capabilities of the tested models, our proposed diagnostics allow to evaluate the quality of datasets by assessing how artefact and bias-free they are, and hence the extent to which they can be trusted for evaluating NLI models' language reasoning capabilities. We consider this step highly important for estimating the quality of existing benchmarks and interpreting model results accordingly, and for guiding the development of new datasets addressing inference and reasoning. We make our code and data available in order to promote the adoption of these diagnostic tests and facilitate their application to new datasets.¹

2 Related Work

A well-known problem of NLU evaluation benchmarks is that the proposed tasks are often solvable by simple heuristics (Hewitt and Liang, 2019). This is mainly due to the presence of linguistic biases in the datasets, which make prediction easy (Lai and Hockenmaier, 2014; Poliak et al., 2018). Notably, 90% of the hypotheses that denote a contradiction in the original SNLI dataset (Bowman et al., 2015) contain the verb *sleep* and its variants (*sleeping*, *asleep*) which serve to mark a contrast with an activity described in the premise (e.g., *My sister is playing* → *My sister is sleeping*); while contradictions in SICK (Marelli et al., 2014) are often marked by explicit negation. This latter issue also exists in SNLI and MNLI as spotted by Gururangan et al. (2018), where negation is highly indicative of contradiction, and generic nouns (e.g., *animal*, *something*) of entailment. These grammatical or lexical cues are easily grasped by the models during training and help them correctly predict the relationship between two sentences, but this does not mean that the models are capable of performing this type of reasoning. Notably, due to these annotation artefacts and statistical ir-

¹<https://github.com/Helsinki-NLP/nli-data-sanity-check>

regularities, it is possible even for hypothesis-only NLI models (i.e. models that are fine-tuned only on the hypotheses without access to the premises) to make correct predictions (Poliak et al., 2018).

Recent work shows that state-of-the-art NLU models are not very sensitive to word order which, however, is one of the most important characteristics of a sequence (Pham et al., 2020). Specifically, performance of BERT-based classifiers fine-tuned on GLUE tasks (Wang et al., 2018) remains relatively high after randomly shuffling input words. This is mainly explained by the contribution of each individual word which remains unchanged after its context is shuffled. Superficial cues such as the sentiment of keywords in sentiment analysis, or the word level similarity between sentence pairs in NLI, allow BERT-based models to make correct decisions even when tokens are arranged in random orders, suggesting that many GLUE tasks are not really challenging them to understand the meaning of a sentence.

To the contrary, when simple heuristics do not suffice to solve the NLI task, NLI systems seem to be more prone to breaking. This is for example what happens when swapping the test and training datasets of different benchmarks (i.e. training on one NLI dataset and testing on an other) (Talman and Chatzikiyiakidis, 2019). Wang et al. (2019b) report problems in performance when the premise and the hypothesis are swapped. The idea is that the label of contradicting or neutral pairs should remain the same in the case of a swap, in contrast to entailment pairs where a different label should be proposed after the swap. This would be expected because entailment is a directional relationship, while contradiction is symmetric.² Wang et al. (2019b) test various models with respect to this diagnostic and observe a significant drop in performance (i.e. predicted labels change) when the contradicting and neutral pairs are swapped. The models' behaviour seems more reasonable when these are tested on the swapped entailment pairs, where all but one models correctly predict a different label. In the light of these results, the authors propose the swapping method as a sanity check for NLI models.

The low quality of existing datasets and the impressively high performance of NLI systems, as measured on these benchmarks, have sparked

²More explicitly, for contradiction, the idea is that when $A \rightarrow \neg B$ (i.e. B contradicts A), then, by contraposition, $B \rightarrow \neg A$ also holds (A contradicts B).

a new research direction where the goal is to propose new more challenging and artefact-free datasets. The ANLI dataset, for example, was built precisely with the goal to eliminate annotation artefacts (Nie et al., 2020). The authors claim that this dataset is much less prone to annotation artefacts compared to previous benchmarks, as suggested by the lower prediction accuracy for models fine-tuned on the ANLI hypothesis-only dataset. Although there still seems to be space for improvement (accuracy is around 0.5, i.e. well above chance), the reported findings are promising. Specifically, the performance is lower than on the hypothesis-only SNLI/MNLI datasets, showing that the dataset contains less artefacts that can guide prediction. ANLI is thus a natural candidate to further test our hypotheses, as it claims to remedy for a number of the shortcomings of earlier NLI datasets.

Lessons learnt from previous work on designing reliable linguistic probing tasks (Hewitt and Liang, 2019) and the overfitting problems of NLI models discussed above, demonstrate the importance of systematic sanity checks like the ones we propose in this paper. Our dedicated control tasks specifically allow to determine whether a dataset triggers the models’ reasoning capabilities or, instead, allows them to rely on statistical biases and annotation artefacts for prediction. We use the quality of the predictions made by models fine-tuned and tested on corrupted data as a proxy to evaluate data quality.

3 Datasets

3.1 The Multi-Genre NLI (MNLI) Corpus

We carry out our experiments on the Multi-Genre Natural Language Inference (MNLI) corpus (Williams et al., 2018). MNLI contains 433k human-written sentence pairs labeled as “entailment”, “contradiction” and “neutral”. The corpus includes sentence pairs from ten distinct genres of written and spoken English,³ making it possible to approximate a wide variety of ways in which

³MNLI text genres: Two-sided in person and telephone conversations (FACE-TO-FACE, TELEPHONE); content from public domain government websites (GOVERNMENT); letters from the Indiana Center for Intercultural Communication of Philanthropic Fundraising Discourse (LETTERS); the public report from the National Commission on Terrorist Attacks Upon the United States (9/11); non-fiction works on the textile industry and child development (OUP); popular culture articles (SLATE); travel guides (TRAVEL); short posts about linguistics for non-specialists (VERBATIM); FICTION.

modern standard American English is used, and supplying a setting for evaluating cross-genre domain adaptation. All ten genres appear in the test and development sets, but only five are included in the training set. The MNLI development and test sets have been divided into “matched” and “mismatched”: The former includes only sentences from the same genres found in the training data, and the latter includes sentences from the remaining genres not present in the training data. For our experiments, we use the development sets as our evaluation data since the annotated test sets are not publicly available.

3.2 The Adversarial NLI (ANLI) Corpus

The Adversarial NLI benchmark (ANLI) (Nie et al., 2020) was specifically designed to address some of the shortcomings of the previous NLI datasets. ANLI contains three datasets (rounds), R1, R2 and R3. Each dataset was collected using a human-and-model-in-the-loop approach, and they progressively increase in difficulty and complexity. The annotators were shown a context (premise) and a target label, and were asked to propose a hypothesis that would lead a model to miss-classify the label. For R1, the model that the annotators were asked to deceive was BERT-Large, while for R2 and R3, it was RoBERTa. For R3, the contexts were selected from a wider set of sources.⁴ The corpus also includes label explanations provided by the annotators. Each round (R1-R3) contains training, development and test data.

ANLI is a relatively small dataset. R1 consists of only 16,946 training examples, 1,000 development and 1,000 test examples. R2 is slightly larger, it contains 45,460 training examples and the same number of development and test examples as R1. Finally, R3 contains 100,459 training examples and slightly larger development and test sets (1,200 each).

3.3 Systematic NLI Data Corruption

We create modified versions of the MNLI training and evaluation data by applying a set of controlled transformations to the original dataset. We call these two sets MNLI CORRUPT-TRAIN and CORRUPT-TEST, respectively. We specifically re-

⁴The contexts for R1 and R2 consist of sentences retrieved from Wikipedia. In R3 the contexts are retrieved from Wikipedia, News (Common Crawl), fiction, The Children’s Book Test (CBT), formal spoken text and procedural text extracted from WikiHow.

move words of specific word classes after tagging the texts with universal part of speech (POS) tags using the NLTK library and the averaged perceptron tagger.⁵ In the obtained MNLI-NOUN training dataset, for example, all nouns in the original MNLI training data have been removed. We furthermore create training data following the inverse process, i.e. keeping only words of specific classes and removing the others. For example, the NOUN+VERB dataset contains only nouns and verbs from the original MNLI sentences.

We similarly create the CORRUPT-TEST set by removing words of specific word classes from the MNLI-matched development dataset, or keeping these and removing the rest. Table 2 in the Appendix contains statistics about the training and evaluation datasets obtained after applying each transformation. Finally, we combine the original MNLI and the corrupted training datasets together. MNLI-ALLDROP contains the following training sets: MNLI (original), -NUM, -CONJ, -ADV, -PRON, -ADJ, -DET, -VERB, -NOUN.

We use ANLI as an example of a high quality dataset, and create ANLI-CORRUPT-TEST by applying all the -POS transformations on the ANLI test sets. Table 3 in the Appendix contains statistics about the different ANLI-CORRUPT-TEST datasets. To test the effect of corrupting the training data used in ANLI experiments (Nie et al., 2020), we also create a training set that consists of the SNLI, MNLI, FEVER and ANLI training data with all the occurrences of nouns removed (ANLI-CORRUPT-TRAIN).

We test the performance of BERT on the corrupted MNLI data, and that of RoBERTa on the corrupted ANLI data, and compare the results to those obtained using the original datasets. We expect models fine-tuned on corrupted data – where important information is missing and sentences often do not make sense – to perform poorly compared to the same models fine-tuned on the original data. High performance of models fine-tuned on these highly problematic data would indicate that the models leverage clues (biases and artefacts) that are present in the data, instead of performing reasoning operations. Inversely, low model performance would suggest that they are unable to reason using these corrupted data, and that the data do not contain artefacts that would guide prediction in this setting.

⁵<https://www.nltk.org/>.

4 Models

We use Google’s original TensorFlow implementation⁶ of the uncased 768-dimensional BERT model (BERT-base), a transformer model that learns representations via a bidirectional encoder (Devlin et al., 2019). BERT was pre-trained using a Masked Language Model (MLM or cloze) task where some percentage of the input tokens are masked at random, and the model needs to predict these masked tokens; and on a Next Sentence Prediction (NSP) task, where it receives pairs of sentences(A, B) as input and learns to predict if B follows A in the original document. Sentence B in (A, B) is 50% of the time the actual sentence that follows A, and 50% of the time it is a random sentence from the training corpus. NSP increases the model’s ability to capture the relationship between two sentences, which is the core task in NLI and Question Answering.

Variants of the BERT model achieve very high performance on NLU tasks, surpassing the human baseline on GLUE (Wang et al., 2018) and reaching near-human performance on the challenging SuperGLUE dataset (Wang et al., 2019a). For each experiment, we fine-tune BERT for ten epochs on the original MNLI training dataset or its transformed versions described in Section 3, using a batch size of 100 (unless explicitly stated).

For the experiments on the ANLI benchmark, we apply the RoBERTa-large model, a variant of BERT which has much higher performance than BERT on the GLUE and SuperGLUE benchmarks.⁷ We use the training and evaluation scripts provided by Nie et al. (2020).⁸ We fine-tune the model for two epochs using a batch size of 16.

5 Evaluation

5.1 CORRUPT-TRAIN and Original Test

We evaluate the performance of the BERT model when fine-tuned on each of the 14 training sets in MNLI CORRUPT-TRAIN. We measure the models’ prediction accuracy on the original MNLI-

⁶<https://github.com/google-research/bert>

⁷The modifications in RoBERTa include training the model longer, with bigger batches, over more data and on longer sequences. The pre-training approaches has also been modified compared to BERT: The next sentence prediction objective is removed and dynamic masking is introduced. This results in different tokens being masked across training epochs.

⁸<https://github.com/facebookresearch/anli>

Data	CORRUPT-TRAIN	Δ	CORRUPT-TEST	Δ	CORRUPT-TRAIN AND TEST	Δ
MNLI-NUM	82.37%	-1.37	81.71%	-2.03	81.87%	-1.87
MNLI-CONJ	83.09%	-0.65	82.75%	-0.99	83.10%	-0.64
MNLI-ADV	80.21%	-3.53	72.41%	-11.33	75.69%	-8.05
MNLI-PRON	83.27%	-0.47	81.98%	-1.75	82.65%	-1.09
MNLI-ADJ	81.67%	-2.07	74.61%	-9.13	76.44%	-7.30
MNLI-DET	83.15%	-0.59	79.29%	-4.44	81.32%	-2.42
MNLI-VERB	81.40%	-2.34	73.96%	-9.78	76.30%	-7.44
MNLI-NOUN	80.72%	-3.02	69.80%	-13.94	73.38%	-10.35
MNLI-NOUN-PRON	79.74%	-4.00	68.41%	-15.33	72.14%	-11.60
NOUN+PRON+VERB	72.55%	-11.19	54.59%	-29.15	62.18%	-21.56
NOUN+ADV+VERB	67.58%	-16.16	62.58%	-21.16	67.58%	-16.16
NOUN+VERB	71.14%	-12.60	52.90%	-30.84	61.31%	-22.43
NOUN+VERB+ADJ	75.54%	-8.20	61.90%	-21.84	68.20%	-15.54
NOUN+VERB+ADV+ADJ	79.81%	-3.93	71.81%	-11.93	76.29%	-7.45

Table 2: Prediction accuracy (%) for the BERT_{base} model fine-tuned on CORRUPT-TRAIN and tested on the original MNLI-matched evaluation (dev) set (columns 2 and 3); fine-tuned on the original MNLI data and tested on CORRUPT-TEST; fine-tuned on CORRUPT-TRAIN and tested on CORRUPT-TEST (columns 6 and 7). The delta shows the difference in accuracy compared to the model fine-tuned on the original MNLI training set and evaluated on the MNLI-matched development set (83.74%).

matched development dataset, which serves as our test set. The results given in the first column of Table 2 show that removing all the occurrences of a specific word class from the MNLI training data has a surprisingly low impact on BERT’s performance, which remains high. As expected, the biggest decrease is observed when content words are removed, with adverbs having the largest impact (-3.53), followed by nouns (-3.02) and verbs (-2.34). Interestingly, the number of nouns is 4.5 times higher than the number of adverbs in the dataset, suggesting that the latter have a larger impact on NLI prediction. The small drop in accuracy observed across the board is, however, highly surprising. Arguably, sentences with nouns removed make very little sense to humans (cf. Table 1).⁹ The observed high performance of BERT on these problematic data might be due to the knowledge about gap filling and Next Sentence Prediction acquired by the model during pre-training, which it can still leverage and combine with other cues in the training and test data for prediction.

5.2 Evaluation on CORRUPT-TEST

Models fine-tuned on original data. We evaluate the performance of the BERT model fine-tuned on the original MNLI training data, on our CORRUPT-TEST data. The middle columns of Table 2 show the experimental results on the different CORRUPT-TEST datasets, and the difference (delta) from the results on the original (unmodi-

⁹Cf. Table 1 in the Appendix for examples of corrupted sentence pairs from the MNLI-NOUN test set for which BERT has made a correct prediction.

fied) MNLI-matched development set.

We observe a similar pattern as in the previous experiment. Removing content words (nouns, verbs and adverbs) has the strongest impact on model accuracy, whereas eliminating conjunctions and numerals has only a small impact on the results. The decrease in prediction accuracy observed in this setting is more important than in the evaluation of models fine-tuned on CORRUPT-TRAIN and tested on unmodified data. Nevertheless, the fact that BERT can still predict the correct label with fairly high accuracy in cases where all the nouns or verbs are removed is surprising, since these transformations often lead to almost unintelligible sentence pairs (cf. examples in Table 1 in the paper and Table 1 in the Appendix). Since inference in such non-sensical sentences cannot rely on meaning, our explanation for the models’ performance is that they leverage other clues and biases that remain in the sentences after corruption for prediction. Note that the models tested in this setting were fine-tuned on the original MNLI data. We believe that during this stage the model acquires knowledge about possible sequence pairs, including the artefacts and other clues therein.

Models fine-tuned on CORRUPT-TRAIN. We evaluate the performance of BERT models fine-tuned on CORRUPT-TRAIN, on CORRUPT-TEST. The results of these experiments are shown in the last two columns of Table 2. We observe again a similar pattern in terms of relative importance of the different word classes, with content words having the biggest impact. What is definitely

Training Data	MNLI-matched (dev)	MNLI-mismatched (dev)
MNLI	83.74%	83.76%
MNLI-ALLDROP	84.09%	84.30%

Table 3: Comparison of prediction accuracy (%) for BERT-base models fine-tuned on the original MNLI training set and on MNLI-ALLDROP, and tested on the original MNLI evaluation (dev) sets.

Data	CORRUPT-TEST R1	Δ	CORRUPT-TEST R2	Δ	CORRUPT-TEST R3	Δ
ANLI-CONJ	70.2%	-3.6	49.0%	0.1	46.5%	2.1
ANLI-PRON	69.6%	-4.2	49.7%	0.8	45.0%	0.6
ANLI-DET	69.5%	-4.3	49.4%	0.5	45.0%	0.6
ANLI-ADV	67.1%	-6.7	49.6%	0.7	43.8%	-0.6
ANLI-ADJ	60.2%	-13.6	45.1%	-3.8	45.0%	0.6
ANLI-NUM	58.7%	-15.1	43.8%	-5.1	45.1%	0.7
ANLI-VERB	54.6%	-19.2	44.7%	-4.2	39.3%	-5.1
ANLI-NOUN	43.7%	-30.1	36.0%	-12.9	32.4%	-12.0

Table 4: Prediction accuracy (%) for the RoBERTa-large model on the CORRUPT R1, R2 and R3 test sets. Delta shows the difference in accuracy compared to the state-of-the-art results reported by Nie et al. (2020) on the original test sets, R1: 73.8%, R2: 48.9% and R3: 44.4%.

surprising in this case is that the drop in performance is smaller than the one observed for the models trained on the original data and tested on CORRUPT-TEST, suggesting that the model relies on data artefacts even more in this setting.

5.3 MNLI-ALLDROP Evaluation

Motivated by the small decrease in prediction accuracy observed when removing specific word classes from the training data (cf. Section 5.1), we also fine-tune the model on a large dataset combining the different CORRUPT-TRAIN sets and the original MNLI training set. The BERT fine-tuning code is shuffling the provided examples, so our goal here is to explore whether seeing sentence pairs where words of different classes are missing (e.g., sentences without verbs following sentences that contain no nouns) confuses the model.

The results of this experiment are shown in Table 3. They indicate that removing occurrences of different word classes from the sentences during training can act as a regularisation technique and, hence improve the model performance. We observe a small increase (+0.35) when evaluated on the original MNLI-matched development data, and an increase of 0.56 when evaluated on the original MNLI-mismatched development data.

5.4 Evaluating on ANLI

In order to demonstrate that systematic data corruption can be a useful diagnostic for evaluating benchmark quality, we conduct additional experiments on the ANLI test set (Nie et al., 2020). The results for the RoBERTa-large model fine-

tuned on the original training sets and evaluated on CORRUPT-TEST R1, R2 and R3 data are given in Table 4.

As expected, we observe a clear drop in accuracy for the datasets where content-bearing words are removed (-NOUNS, -VERBS), and a relatively small drop when function words are missing (-CONJ, -DET), but only in R1. However, the fact that accuracy on the R2 and R3 datasets improves after some corruption transformations are applied (ANLI-PRON, -CONJ, -DET) is an interesting finding. A possible explanation is that as the sentences (especially the premises) are much longer in ANLI compared to other NLI datasets, removing non-content-bearing words makes it easier for the model to grasp the essential information for making correct predictions. The large drop in accuracy when nouns and verbs are removed supports our hypothesis regarding the superior quality of the ANLI corpus compared to MNLI, suggesting that the dataset contains less artefacts on which the model can base prediction after corruption.

We also compare the results reported by Nie et al. (2020) for the RoBERTa-large model to the ones obtained with the model fine-tuned on the ANLI-NOUN training set.¹⁰ We measure the model’s prediction accuracy on the original R1, R2 and R3 test sets, and report the results in Table 5. The drop in prediction accuracy is significantly larger than that observed on the MNLI data. Hence, the data corruption procedure reveals the

¹⁰This corresponds to MNLI+SNLI+FEVER+ANLI with all nouns removed.

Training data	R1	R2	R3
ANLI	73.8%	48.9%	44.4%
ANLI-NOUN	57.6%	40.3%	41.0%

Table 5: Prediction accuracy (%) for RoBERTa-large on the ANLI-NOUN dataset. Comparison to the results of Nie et al. (2020) on the original ANLI dataset. ANLI contains MNLI, SNLI, FEVER and ANLI.

improved quality of the ANLI data set as a benchmark for NLU. However, the fact that the model is able to predict the correct label with 57.6% accuracy for ANLI R1 highlights that even with this dataset the model learns some factors from the data that it is able to use when predicting the label for a pair, even when the training sentences do not make much sense. These results further demonstrate the importance of carefully running diagnostics such as ours to assess the use of a new benchmark in NLU tasks.

6 Discussion

The question of whether current state-of-the-art neural network models that beat human performance in NLU tasks actually understand language is currently much debated. Our proposed corruption transformations often lead to sentences that make very little sense. Nevertheless, we observe that BERT performs surprisingly well in these experiments. This indicates that rather than understanding the meaning of the sentences and the semantic relationship between them, the models are able to pick up on other cues from the data that allow them to make correct predictions.

Our proposed diagnostics tests are useful devices for assessing the quality of a dataset as a testbed for evaluating models’ language understanding capabilities. In our experiments, they demonstrate the superior quality of a NLI dataset (ANLI) over another (MNLI). We test this finding in an additional experiment where we apply the word shuffling mechanism of Pham et al. (2020) on the ANLI data, which was shown to not deteriorate BERT-based model performance on the GLUE tasks. Our results in Table 6 show that this procedure significantly hurts model accuracy on ANLI, and bring in additional evidence supporting the superior quality of this dataset over MNLI (which is part of the GLUE benchmark).

Our test suite can be seen as an additional “crash test” for assessing the quality of bench-

Test set	R1	R2	R3
ANLI	73.8%	48.9%	44.4%
ANLI-SHUFFLE-n1	35.5%	33.8%	36.0%
ANLI-SHUFFLE-n2	45.4%	39.8%	37.1%
ANLI-SHUFFLE-n3	49.4%	40.7%	38.4%

Table 6: Prediction accuracy (%) for RoBERTa-large after word shuffling (Pham et al., 2020). Comparison to results obtained on the original ANLI dataset (Nie et al., 2020). The ANLI-SHUFFLE-n1/n2/n3 test sets contain shuffled n-grams, with $n = \{1, 2, 3\}$ respectively.

mark datasets that address common-sense reasoning. It falls in the same line as work that highlighted problems of earlier datasets and resulted in the creation of ANLI. Our proposition can be part of a good methodology for building future NLI datasets. The multi-faceted nature of the problems that exist in current NLI datasets makes research that investigates these issues very important; the more the diagnostic tests we have, the more reliable the datasets will hopefully get. The fact that one type of testing (hypothesis only, word shuffling or word class dropping) does not eliminate all problems present in the datasets, highlights the need for a variety of diagnostic devices addressing different phenomena.

We propose the following set of diagnostics as the minimum sanity check when developing new NLI datasets:

- Hypothesis only baseline (Gururangan et al., 2018; Poliak et al., 2018)
- Word-order shuffling (Pham et al., 2020)
- Swapping premises and hypotheses (Wang et al., 2019b)
- Word class dropping (our proposed diagnostics)

Returning to the specific findings of this paper, we performed an additional set of analysis aimed at identifying what the observed, relatively small, impact of the proposed modifications is due to. We explore whether the drop in performance can be explained by the (smaller or larger) number of tokens pertaining to the word class being removed. As can be seen in Figure 1, where we compare the accuracy of BERT and the number of tokens removed from the training data in each setting, this factor does not explain the obtained results. For example, there are only 492,895 occur-

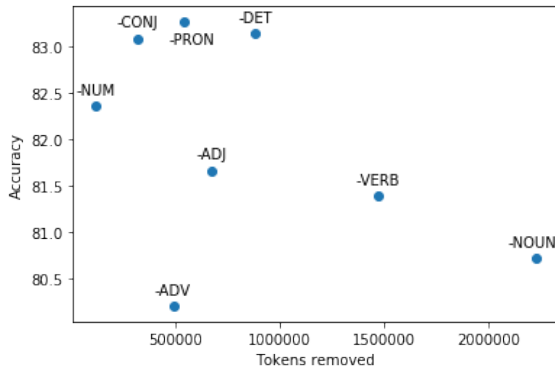


Figure 1: Comparison of BERT-base model Accuracy vs Tokens removed. The model is fine-tuned on the MNLI training data with instances of a specific word class removed, and evaluated on the original MNLI-matched development data.

removals of adverbs removed from the training set, but the delta to the original result is the highest (-3.53 points), whereas removing 886,966 determiners has only a small impact on accuracy (-0.59 points). This plot demonstrates the important role of content words in NLI prediction.

Zhou and Bansal (2020) have shown that high lexical overlap between premises and hypotheses can guide models’ predictions. We thus explore the extent to which our results can be explained by the amount of lexical overlap in the CORRUPT-TEST sets. We measure lexical overlap by counting the tokens shared by the premise and the hypothesis in a sentence pair. The orange bars in the plot in Figure 2 show the amount of lexical overlap between premises and hypotheses (% calculated over the total number of examples) in the original MNLI and the CORRUPT-TEST test sets. The blue bars show the prediction accuracy obtained by BERT fine-tuned on the original MNLI data when evaluated on each test set. We observe that although there is a decrease in lexical overlap in some test sets (e.g., in MNLI-NOUN), there is no clear correlation between lexical overlap and accuracy, which suggests that the model picks up on other cues that remain in the corrupted sentences for prediction.

7 Conclusion

We propose a novel diagnostics suite for assessing the quality of datasets used for NLI model training and evaluation. We show that data corruption is an efficient way to estimate dataset quality and their

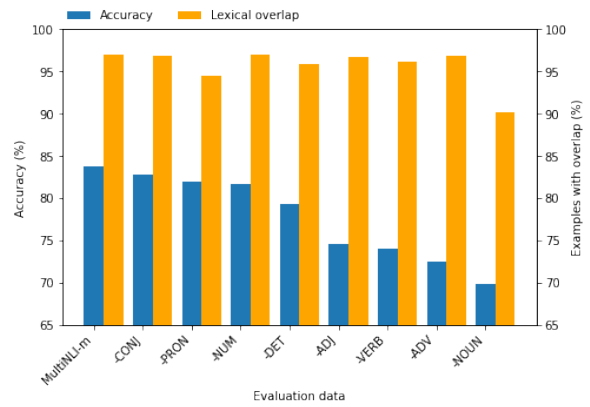


Figure 2: Comparison of model accuracy and lexical overlap in the original MNLI test and the CORRUPT-TEST sets. The models are fine-tuned on the original MNLI training data.

potential to reflect the real language understanding capabilities of the models. Our results on the MNLI and ANLI datasets show that our methodology can successfully identify datasets of high or low quality, i.e. whether a dataset triggers models’ reasoning potential or rather allows them to rely on cues and other statistical biases for prediction. Our proposed tests can be used for assessing the quality of existing benchmarks used by the community and interpreting the results accordingly, and also to guide the development of new datasets addressing reasoning tasks. In this latter case, data corruption would serve to identify whether a dataset construction methodology and the adopted annotation guidelines are on the correct track.

Lastly, although it would be interesting to compare a larger number of architectures, we leave this comparison for future work due to lack of space and also in order to not confuse the reader, given the large number of settings where experiments are conducted. We also focus in this paper on the MNLI and ANLI datasets, since our main concern is to cover as many corruption settings as possible. Extending the current work to other models and NLU datasets is a natural next step for future research. We have made our code available to promote research in this direction.¹¹ Additionally, since the present work leaves open questions as regards the factors behind the high performance observed on the corrupted datasets, we plan to more thoroughly investigate the cues and artefacts on which the models rely and which allow them to

¹¹<https://github.com/Helsinki-NLP/nli-data-sanity-check>

perform well in these tasks.

Acknowledgments



Marianna Apidianaki and Jörg Tiedemann are supported by the FoTran project, funded by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement no. 771113). Stergios Chatzikiyriakidis is supported by grant 2014-39 from the Swedish Research Council, which funds the Centre for Linguistic Theory and Studies in Probability (CLASP) in the Department of Philosophy, Linguistics, and Theory of Science at the University of Gothenburg. We thank the reviewers for their thoughtful comments and valuable suggestions.

References

- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. <https://doi.org/10.18653/v1/D15-1075> A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*, pages 632–642, Lisbon, Portugal.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. <https://doi.org/10.18653/v1/N19-1423> BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of ACL*.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation Artifacts in Natural Language Inference Data. In *Proceedings of NAACL: HLT*, pages 107–112, New Orleans, Louisiana.
- John Hewitt and Percy Liang. 2019. Designing and Interpreting Probes with Control Tasks. In *Proceedings of EMNLP-IJCNLP*, pages 2733–2743, Hong Kong, China.
- Alice Lai and Julia Hockenmaier. 2014. <https://doi.org/10.3115/v1/S14-2055> Illinois-LH: A Denotational and Distributional Approach to Semantics. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 329–334, Dublin, Ireland.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. <http://arxiv.org/abs/1907.11692> Roberta: A robustly optimized bert pretraining approach.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC*, pages 216–223, Reykjavik, Iceland.
- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. <https://doi.org/10.18653/v1/2020.acl-main.441> Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.
- Thang M. Pham, Trung Bui, Long Mai, and Anh Nguyen. 2020. Out of Order: How important is the sequential order of words in a sentence in Natural Language Understanding tasks? *arXiv preprint arXiv:2012.15180*.
- Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*.
- Aarne Talman and Stergios Chatzikiyriakidis. 2019. Testing the Generalization Power of Neural Network Models across NLI Benchmarks. In *Proceedings of BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 85–94, Florence, Italy.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32, pages 3266–3280. Curran Associates, Inc.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. <https://doi.org/10.18653/v1/W18-5446> GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium.
- Haohan Wang, Da Sun, and Eric P Xing. 2019b. What if we simply swap the two text fragments? a straightforward yet effective way to test the robustness of methods to confounding signals in nature language

inference tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7136–7143.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL*.

Xiang Zhou and Mohit Bansal. 2020. Towards robustifying NLI models against lexical dataset biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8759–8771. Association for Computational Linguistics.

Appendix

Table 1 contains examples of sentence pairs from the MNLI-NOUN test set for which BERT predicted the correct labels. Table 2 contains statistics for the number of tokens removed from the corrupted MNLI datasets. Table 3 contains statistics for the number of tokens removed from the corrupted ANLI test sets.

Label	Premise	Hypothesis
contradiction	<i>The intends that with appropriate in developing this.</i>	<i>The discourages to consult with any.</i>
contradiction	<i>Like and, warns, and Japanese are joined by yet locked in traditional.</i>	<i>and Japanese have no between them.</i>
contradiction	<i>To be sure, not all are.</i>	<i>Every single is a.</i>
entailment	<i>The, or Where the?</i>	<i>The of saving.</i>
entailment	<i>In the original, is set up by his and then ambushed by a hostile named, and when he tries to answer with an eloquent (is clenched.</i>	<i>is out to get him.</i>
entailment	<i>The other is retrospective and intended to help those who review to assess the of completed.</i>	<i>It is made to help the assess the of the.</i>
neutral	<i>and uh it that takes so much away from your</i>	<i>you away from your because it is more important to you.</i>
neutral	<i>The had been found in a in the.</i>	<i>The that was in the was powdered.</i>
neutral	<i>In the other, the beat the.</i>	<i>The are a better.</i>

Table 1: Randomly selected sentence pairs from MNLI-NOUN test set for which BERT predicted the correct labels.

Configuration	Training datasets			Test datasets		
	Premises	Hypotheses	Total	Premises	Hypotheses	Total
MNLI-NUM	119,587	44,289	163,876	3,100	1,133	4,233
MNLI-CONJ	320,210	76,466	396,676	7,584	1,874	9,458
MNLI-ADV	492,895	237,250	730,145	11,777	5,862	17,639
MNLI-PRON	543,968	301,293	845,261	13,060	7,466	20,526
MNLI-ADJ	677,095	302,652	979,747	16,162	7,562	23,724
MNLI-DET	886,966	483,238	1,370,204	21,198	11,723	32,921
MNLI-VERB	1,474,454	886,597	2,361,051	35,813	22,101	57,914
MNLI-NOUN	2,228,780	1,090,814	3,319,594	54,700	27,182	81,882
MNLI-NOUN-PRON	2,772,748	1,392,107	4,164,855	67,760	34,648	102,408
NOUN+PRON+VERB	4,501,189	2,166,146	6,667,335	109,325	53,647	162,972
NOUN+ADV+VERB	4,552,262	2,230,189	6,782,451	110,608	55,251	165,859
NOUN+VERB	5,045,157	2,467,439	7,512,596	122,385	61,113	183,498
NOUN+VERB+ADJ	4,368,062	2,164,787	6,532,849	106,223	53,551	159,774
NOUN+VERB+ADV+ADJ	3,875,167	1,927,537	5,802,704	94,446	47,689	142,135

Table 2: Datasets formed by removing tokens from MNLI. The numbers correspond to number of tokens removed from the Premises and Hypotheses, and the total number of removed tokens.

Test dataset	R1			R2			R3		
	Premises	Hypotheses	Total	Premises	Hypotheses	Total	Premises	Hypotheses	Total
ANLI-NOUN	23,523	4,719	28,242	23,646	4,275	27,921	23,086	4,033	27,119
ANLI-VERB	6,057	1,657	7,714	6,155	1,668	7,823	11,281	2,258	13,539
ANLI-PRON	1,567	184	1,751	1,657	178	1,835	4,152	446	4,598
ANLI-ADJ	2,827	514	3,341	2,783	495	3,278	3,525	625	4,150
ANLI-ADV	899	267	1,166	917	313	1,230	2,898	470	3,368
ANLI-NUM	2,934	576	3,510	2,862	515	3,377	1,737	286	2,023
ANLI-CONJ	1,816	161	1,977	1,897	122	2,019	2,073	142	2,215
ANLI-DET	5,631	1,195	6,826	5,669	1,086	6,755	7,167	1,406	8,573

Table 3: Datasets formed by removing tokens from ANLI test sets. The numbers correspond to number of tokens removed from the Premises and Hypotheses, and the total number of removed tokens for the three datasets (rounds).

Finnish Paraphrase Corpus

Jenna Kanerva, Filip Ginter, Li-Hsin Chang, Iiro Rastas, Valtteri Skantsi,
Jemina Kilpeläinen, Hanna-Mari Kupari, Jenna Saarni, Maija Sevón, and Otto Tarkka

TurkuNLP group
Department of Computing
Faculty of Technology
University of Turku, Finland
jmnybl@utu.fi

Abstract

In this paper, we introduce the first fully manually annotated paraphrase corpus for Finnish containing 53,572 paraphrase pairs harvested from alternative subtitles and news headings. Out of all paraphrase pairs in our corpus 98% are manually classified to be paraphrases at least in their given context, if not in all contexts. Additionally, we establish a manual candidate selection method and demonstrate its feasibility in high quality paraphrase selection in terms of both cost and quality.

1 Introduction

The powerful language models that have recently become available in NLP have also resulted in a distinct shift towards more meaning-oriented tasks for model fine-tuning and evaluation. The most typical example is entailment detection, with the paraphrase task raising in interest recently. Paraphrases, texts that express the same meaning with differing words (Bhagat and Hovy, 2013), are — already by their very definition — a suitable target to induce and evaluate models’ ability to represent meaning. Paraphrase detection and generation has numerous direct applications in NLP (Madnani and Dorr, 2010), among others in question answering (Soni and Roberts, 2019), plagiarism detection (Altheneyan and Menai, 2019), and machine translation (Mehdizadeh Seraj et al., 2015).

Research in paraphrase naturally depends on the availability of datasets for the task. We will review these in more detail in Section 2, nevertheless, barring few exceptions, paraphrase corpora are typically large and gathered automatically using one of several possible heuristics. Typically a comparatively small section of the corpus is manually classified to serve as a test set for method development. The heuristics used to gather and filter the

corpora naturally introduce a bias to the corpora which, as we will show later in this paper, demonstrates itself as a tendency towards short examples with a relatively high lexical overlap. Addressing this bias to the extent possible, and providing a corpus with longer, lexically more diverse paraphrases is one of the motivations for our work. The other motivation is to cater for the needs of Finnish NLP, and improve the availability of high-quality, manually annotated paraphrase data specifically for the Finnish language.

In this paper, we therefore aim for the following contributions: Firstly, we establish and test a fully manual procedure for paraphrase candidate selection with the aim of avoiding a selection bias towards short, lexically overlapping candidates. Secondly, we release the first fully manually annotated paraphrase corpus of Finnish, sufficiently large for model training. The number of manually annotated examples makes the released dataset one of the largest, if not the largest manually annotated paraphrase corpus for any language. And thirdly, we report the experiences, tools, and baseline results on this new dataset, hopefully allowing other language NLP communities to assess the potential of developing a similar corpus for other languages.

2 Related Work

Statistics of the different paraphrase corpora most relevant to our work are summarized in Table 1. For English, the **Microsoft Research Paraphrase Corpus (MRPC)** (Dolan and Brockett, 2005) is extracted from an online news collection by applying heuristics to recognize candidate document pairs and candidate sentences from the documents. Paraphrase candidates are subsequently filtered using a classifier, before the final manual binary annotation (paraphrase or not). In the **Twitter URL Corpus (TUC)** (Lan et al., 2017), paraphrase candidates are identified by recognizing

Corpus	Data source	Size autom.	Size manual	Labels
English				
MRPC	Online news	—	5,801	0/1
TUC	News tweets	—	52K	0/1
ParaSCI	Scientific papers	350K	—	1-5
PARADE	flashcards (computer sci.)	—	10K	0-3
QQP	Quora	404K	—	0/1
Finnish				
Opusparcus	OpenSubtitles	480K*	3,703	1-4
TaPaCo	Tatoeba crowdsourcing	12K	—	—

Table 1: Summary of available paraphrase corpora of naturally occurring sentential paraphrases. The corpora sizes include the total amount of pairs in the corpus (i.e. also those labeled as non-paraphrases), thus the actual number of good paraphrases depend on the class distribution of each corpus. *The highest quality cutpoint estimated by the authors.

shared URLs in news related tweets. All candidates are manually binary-labeled. **ParaSCI** (Dong et al., 2021) is created by collecting paraphrase candidates from ACL and arXiv papers using heuristics based on term definitions, citation information as well as sentence embedding similarity. The extracted candidates are automatically filtered, but no manually annotated data is available. **PARADE** (He et al., 2020) is created by collecting online user-generated flashcards for computer science related concepts. All definitions for the same term are first clustered, and paraphrase candidates are extracted only among a cluster to reduce noise in candidate selection. All extracted candidates are manually annotated using a scheme with four labels. **Quora Question Pairs (QQP)**¹ contains question headings from the forum with binary labels into duplicate-or-not questions. The QQP dataset is larger than other datasets, however, although including human-produced labels, the labeling is not originally designed for paraphrasing and the dataset providers warn about labeling not guaranteed to be perfect.

Another common approach for automatic paraphrase identification is through language pivoting using multilingual parallel datasets. Here sentence alignments are used to recognize whether two different surface realizations share an identical or near-identical translation, assuming that the identical translation likely implies a paraphrase. There are two different multilingual paraphrase datasets automatically extracted using language pivoting, **Opusparcus** (Creutz, 2018) and **TaPaCo** (Scherrer,

2020), both including a Finnish subsection. **Opusparcus** consists of candidate paraphrases automatically extracted from the alternative translations of movie and TV show subtitles after automatic sentence alignment. While the candidate paraphrases are automatically extracted, a small subset of a few thousand paraphrase pairs for each language is manually annotated. **TaPaCo** contains candidate paraphrases automatically extracted from the Tatoeba dataset², which is a multilingual crowdsourced database of sentences and their translations. Like Opusparcus, TaPaCo is based on language pivoting, where all alternative translations for the same statement are collected. However, unlike most other corpora, the candidate paraphrases are grouped into ‘sets’ instead of pairs, and all sentences in a set are considered equivalent in meaning. TaPaCo does not include any manual validation.

3 Text Selection

As discussed previously, we elect to rely on fully manual candidate extraction as a measure against any bias introduced through heuristic candidate selection methods. In order to obtain sufficiently many paraphrases for the person-months spent, the text sources need to be paraphrase-rich, i.e. have a high probability for naturally occurring paraphrases. Such text sources include for example news headings and articles reporting on the same news, alternative translations of the same source material, different student essays and exam answers for the same assignment, and related questions with their replies in discussion fora, where

¹data.quora.com/First-Quora-Dataset-Release-Question-Pairs

²<https://tatoeba.org/eng/>

one can assume different writers using distinct words to state similar meaning. For this first version of the corpus, we use two different text sources: alternative Finnish subtitles for the same movies or TV episodes, and headings from news articles discussing the same event in two different Finnish news sites.

3.1 Alternative Subtitles

OpenSubtitles³ distributes an extensive collection of user generated subtitles for different movies and TV episodes. These subtitles are available in multiple languages, but surprisingly often the same movie or episode have versions in a single language, originating from different sources. This gives an opportunity to exploit the natural variation produced by independent translators, and by comparing two different subtitles for a single movie or episode, there is a high likelihood of finding naturally occurring paraphrases.

From the database dump of OpenSubtitles2018 obtained through OPUS (Tiedemann, 2012), we selected all movies and TV episodes with at least two Finnish subtitle versions. In case more versions are available, the two most lexically differing are selected for paraphrase extraction. We measure lexical similarity by TF-IDF weighted document vectors. Specifically, we create TF-IDF vectors with `TfidfVectorizer` from the `sklearn` package. We limit the number of features to 200K, apply sublinear scaling, use character 4-grams created out of text inside word boundaries, and otherwise use the default settings. To filter out subtitle pairs with low density of interesting paraphrase candidates, pairs with too high or too low cosine similarity of TF-IDF vectors are discarded. High similarity usually reflects identical subtitles with minor formatting differences, while low similarity is typically caused by incorrect identifiers in the source data. The two selected subtitle versions are then roughly aligned using the timestamps, and divided into segments of 15 minutes. For every movie/episode, the annotators are assigned one random such segment, the two versions presented side-by-side in a custom tool, allowing for fast selection of paraphrase candidates.

In total, we were able to obtain at least one pair of aligned subtitle versions for 1,700 unique movies and TV series. While for each unique movie only one pair of aligned subtitles is se-

lected for annotation, TV series comprise different episodes, dealing with the same plot and characters, and therefore overlapping in language. After an initial annotation period, we noticed a topic bias towards a limited number of TV series with a large number of episodes, and decided to limit the number of annotated episodes to 10 per each TV series in all subsequent annotation. In total, close to 3,000 different movies/episodes are used for manual paraphrase candidate extraction, each including exactly one pair of aligned subtitles.

3.2 News Headings

We have downloaded news articles through open RSS feeds of different Finnish news sites during 2017–2021, resulting in a substantial collection of news from numerous complementary sources. For this present work, we narrow the data down to two sources: the Finnish Broadcasting Company (YLE) and Helsingin Sanomat (HS, English translation: Helsinki News). We align the news using a 7-day sliding window on time of publication, combined with cosine similarity of TF-IDF-weighted document vectors induced on the article body, obtaining article pairs likely reporting on the same event. The settings of the TF-IDF vectors is the same as in Section 3.1. We use the article headings as paraphrase candidates, striving to select maximally dissimilar headings of maximally similar articles as the most promising candidates for non-trivial paraphrases. In practice, we used a simple grid search and human judgement to establish the most promising region of article body and heading similarity values.

4 Paraphrase Annotation

The paraphrase annotation is comprised of multiple annotation steps, including candidate selection as described above, manual classification of candidates based on an annotation scheme, as well as the possibility of rewriting partial paraphrases into full paraphrases. Next, we will discuss the different paraphrase types represented in our annotation scheme, and afterwards the annotation workflow is discussed in a more detailed fashion.

4.1 Annotation Scheme

Instead of a simple yes/no (*equivalent* or *not equivalent*) as in MRPC (Dolan and Brockett, 2005) or 1–4 scale (*bad*, *mostly bad*, *mostly good* and *good*) as in Opusparcus (Creutz, 2018), our

³<http://www.opensubtitles.org>

annotation scheme is adapted to capture the level of paraphrasability in a more detailed fashion. Our annotation scheme uses the base scale 1–4 similar to other paraphrase corpora, enriched with additional subcategories (flags) for distinguishing different types of paraphrases which would otherwise fall from the label 4 (*good*) into label 3 (*mostly good*).

An example for each of the categories discussed below is shown in Table 2 (English translations available in Appendix A). Each candidate pair is first evaluated in terms of the base scale numbered from 1 to 4, where:

Label 4 is a full (perfect) paraphrase in all reasonably imaginable contexts, meaning one can always be replaced with the other without changing the meaning. This ability to substitute one for the other in any context is the primary test for *label 4* used in the annotation.

Label 3 is a context dependent paraphrase, where the meaning of the two statements is the same in the present context, but not necessarily in other contexts.

Label 2 is related but not a paraphrase, where there is a clear relation between the two statements, yet they cannot be considered paraphrases.

Label 1 is unrelated, there being no reasonable relation between the two statements, most likely a false positive in candidate selection.

If labeling a candidate pair is not possible for a reason, or giving a label would not serve the desired purpose (e.g. wrong language or identical statements), the example can be skipped with the *label x*.

With the base labels alone, a great number of candidate paraphrases would fail the substitution test for *label 4* and be classified *label 3*. This is especially so for longer text segments which are less likely to express strictly the same meaning. In order to avoid populating the *label 3* category with a very diverse set of paraphrases, we opt to introduce flags for finer sub-categorization and thus support a broader range of downstream applications of the corpus. These flags are always attached to *label 4* (subcategories of full paraphrases), meaning the paraphrases are not fully interchangeable due to the specified reason, but, crucially, are context-independent, unlike *label 3*. The possible flags are:

Subsumption (> or <) where one of the statements is more detailed and the other more general. The relation of the pair is therefore directional, where the more detailed statement can be replaced with the more general one in all contexts, but not the other way around. The two common cases are one statement having additional minor details the other omits, and one statement being ambiguous while the other not. If there is a justification for crossing directionality (one statement being more detailed in one aspect while the other in another aspect), the pair falls into *label 3* as the directional replacement test does not hold anymore.

Style (s) for tone or register difference in cases where the meaning of the two statements is the same, but the statements differ in tone or register such that in certain situations, they would not be interchangeable. For example, if one statement uses pejorative language or profanities, while the other is neutral, or one is clearly colloquial language while the other is formal. The style flag also includes differences in the level of politeness, uncertainty, and strength of the statements.

Minor deviation (i) marks in most cases minimal differences in meaning (typically "this" vs. "that") as well as easily traceable differences in grammatical number, person, tense or such. Some applications might consider these as *label 4* for all practical purposes (e.g. information retrieval), while others should regard these as *label 2* (e.g. automatic rephrasing).

The flags are independent of each other and can be combined in the annotation.

4.2 Annotation Workflow

Given two aligned documents as described in Section 3, an annotator first extracts all candidate paraphrases. These can be anything between a short phrase and several sentences long, typically being about a sentence long. The annotators are encouraged to select as long continuous statements as possible, nevertheless at the same time avoiding a bias towards subsumption flag by over-extending one of the candidates. The candidate paraphrases are subsequently transferred into a classification annotation tool. In case of news headings, where the candidates are extracted automatically, the candidates are introduced directly in the classification tool without any manual extraction step.

Label	Statement 1	Statement 2
4	Tyrmistyttävän lapsellista!	Pöyristyttävän kypsymätöntä!
4s	Olen työskennellyt lounaan ajan.	Tein töitä koko ruokiksen.
4i	Teitpä onnisti.	Oletpa onnekas.
4>	Tein lujasti töitä niiden rahojen eteen.	Paiskin kovasti töitä.
4<s	Sä ruletat! Anna mennä!	Sinä olet paras, Tähhä! Anna mennä!
4is	Sä pöllit meidän kasvin!	Varastit meidän kasvit!
3	Aion tehdä kokeen.	Aion testata sitä.
2	Tappion kokenut Väyrynen katosi Helsingin yöhön.	Väyrynen putoamassa eduskunnasta.
Rewrites		
Orig	Voinko palata tehtäviini?	Saanko jatkaa?
Rew	<i>Voinko palata tehtäviini?</i>	<i>Saanko jatkaa tehtäviäni?</i>

Table 2: Example paraphrase pairs annotated with different labels and flags (English translations available in Appendix A).

In the classification tool, the annotator assigns a label for each candidate. The candidate paraphrases are shown one pair at a time, and for each pair the document context is available.

In addition to assigning a label and optional flags for a candidate pair, the classification tool provides an option to rewrite the statements if the classification is anything else than *label 4* without any flags. The annotators are instructed to rewrite the candidates in cases, where a simple fix, for example word or phrase deletion, addition or replacement with a synonym or changing an inflection, can be easily constructed. Rewrites must be such that the annotated label for the rewritten example is *4*. In cases where the rewrite would require more complicated changes or would take too much time, the annotators are instructed to move on to the next candidate pair. One rewrite done during the data annotation is illustrated in Table 2.

The annotators can mark unsure, difficult or otherwise interesting cases for later discussion in daily annotation meetings. The annotators also communicate online, for instance seeking a quick validation for a particular decision. The work is further supported by a jointly produced 17-page annotation manual, which is revised and extended regularly.

The annotation work is carried out by 5 annotators each working full-time or part-time throughout the 4 month period used to construct the first release version of the corpus. Each annotator has a strong background in language studies by having an academic degree or ongoing studies in a field related to languages or linguistics.

Section	Examples	Rewrites	Total
Train	36,600	6,239	42,839
Devel	4,474	884	5,358
Test	4,589	786	5,375
Total	45,663	7,909	53,572

Table 3: Data sizes in our corpus.

5 Corpus Statistics and Evaluation

The released corpus includes 45,663 naturally occurring paraphrases with additional 7,909 rewrites, resulting in the total size of 53,572 paraphrase pairs. The data is randomly divided into training, development and test sections using 80/10/10 split, however, with a restriction of all paraphrases from the same movie or TV episode being in the same section. Basic data statistics are summarized in Table 3, and label distribution in Figure 1. Notably, the amount of candidate pairs labeled as not paraphrases (labels 1 or 2 in our scheme) is almost non-existent, owing to the manual candidate selection step in subtitles data from which the vast majority of the corpus data originates. Only 5.6% of paraphrase pairs in the corpus originate from the automated candidate selection from news data. The amount of candidates labeled with label 1 or label x is insignificantly small, therefore we decided to discard these from the final corpus.

In Figure 2 we measure the density of different label combinations in the training set conditioned on cosine similarity of paraphrase pairs based on TF-IDF weighted character n-grams of lengths 2–4. Up to cosine similarity of 0.5 the most common labels are evenly represented, while the prevalence

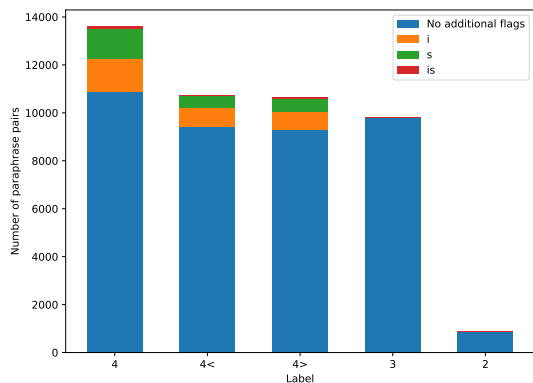


Figure 1: Labels distribution in our corpus excluding 7,909 rewrites which can be added up with *label 4*.

of label 4 increases throughout the range and dominates the sparsely populated range of similarities over 0.8.

5.1 Annotation Quality

After the initial annotator training phase most of the annotation work is carried out as single annotation. In order to monitor annotation consistency, double annotation batches are assigned regularly. In double annotation, one annotator first extracts the candidate paraphrases from the aligned documents, but later on these candidates are assigned to two different annotators, who annotate the labels for these independently from each other. Next, these two individual annotations are merged and conflicting labels are resolved together with the whole annotation team in a meeting. These consensus annotations constitute a gold standard against which individual annotators can be measured.

A total of 1,175 examples are double annotated (2.5% of the data⁴). Most of these are annotated by exactly two annotators, however, some examples may include annotations from more than two annotators, and thus the total amount of individual annotations for which the gold standard label exists is 2,513. We measure the agreement of individually annotated examples against the gold standard annotations in terms of accuracy, i.e. the proportion of individually annotated examples with correctly assigned label.

⁴During the initial annotator training double annotation was used extensively; this annotator training data is not included in the released corpus.

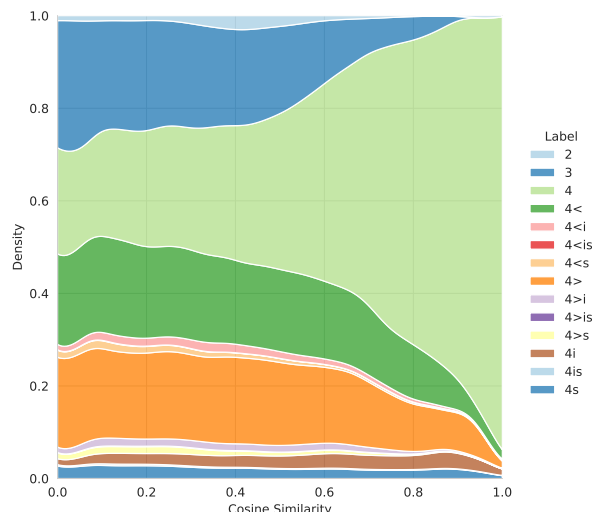


Figure 2: Density of different labels in the training set conditioned on cosine similarity of the paraphrase pairs.

The overall accuracy is 68.7% when the base label (*labels 1–4*) as well as all additional flags are taken into consideration. When discarding the least common flags *s* and *i* and evaluating only base labels and directional subsumption flags, the overall accuracy is 72.9%. To compare the observed agreement to previous studies on paraphrase annotation, the Opusparcus annotation agreement is approximately 64% on Finnish development set and 67% on test set (calculated from numbers in Table 4 and Table 5 in Creutz (2018)). The Opusparcus uses an annotation scheme with four labels, similar to our base label scheme. In MRPC, the reported agreement score is 84% on a binary paraphrase-or-not scheme. While direct comparison is difficult due to the different annotation schemes and label distributions, the figures show that the observed agreement seem to be roughly within the same range with agreement numbers seen in related works.

In addition to agreement accuracy, we calculate two versions of Cohen’s kappa, a metric for inter-annotator agreement taking into account the possibility of agreement occurring by chance. First we measure kappa agreement of all individual annotations against the gold standard, an approach typical in paraphrase literature. This kappa is 0.62, indicating substantial agreement. Additionally, we measure the Cohen’s kappa between each pair of annotators. The weighted average kappa over all annotator pairs is 0.41 indicating moderate agree-

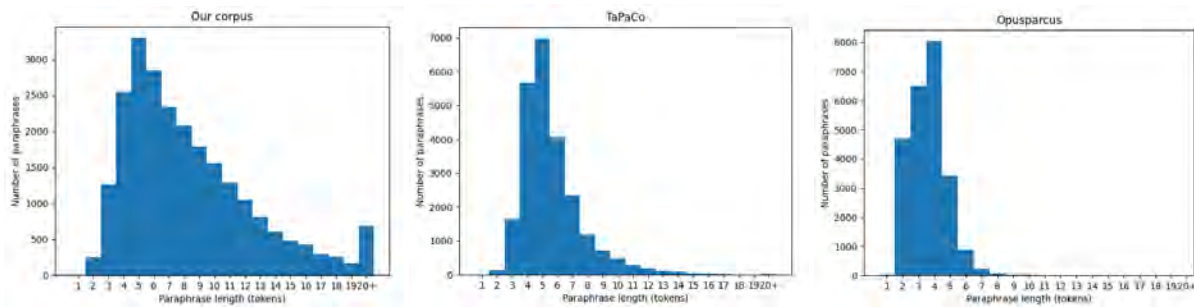


Figure 3: Comparison of paraphrase length distributions in terms of tokens per paraphrase.

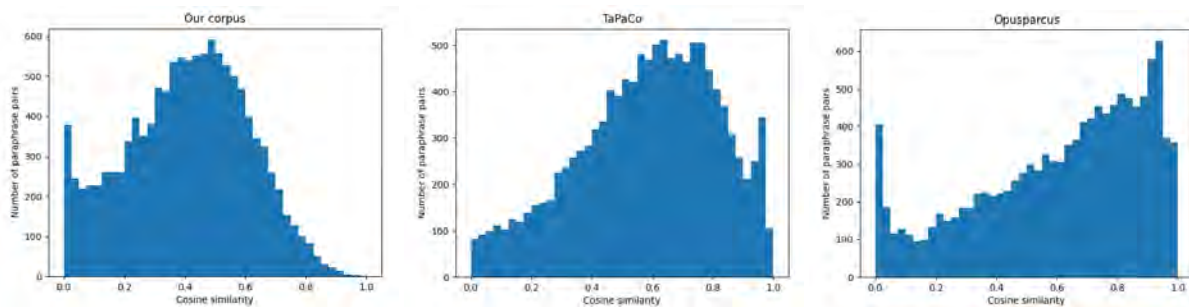


Figure 4: Comparison of paraphrase pair cosine similarity distributions.

ment. Both are measured on full labels. When evaluating only on base labels and directional subsumption flags, these kappa scores are 0.65 and 0.45, respectively.

5.2 Corpus Comparison

We compare the distribution of paraphrase lengths and lexical similarity with the two Finnish paraphrase candidate corpora, Opusparcus and TaPaCo, as the reference. Direct comparison is complicated by several factors. Firstly, both Opusparcus and TaPaCo consist primarily of automatically extracted paraphrase candidates, Opusparcus having only small manually curated development and test sections, and TaPaCo being fully uncurated. Secondly, the small manually annotated sections of Opusparcus are sampled to emphasize lexically dissimilar pairs, and therefore not representative of the characteristics of the whole corpus. We therefore compare with the fully automatically extracted sections of both Opusparcus and TaPaCo. For our corpus, we discard the small proportion of examples of base labels 1 and 2, i.e. not paraphrases. Another important factor to consider is that the proportion of false candidates in the automatically extracted sections of Opusparcus and TaPaCo is unknown, further decreasing comparability: the characteristics of false and true candidates may differ substantially, false candidates for

example likely being on average more dissimilar in terms of lexical overlap than true candidates.

For each corpus, we sample 12,000 paraphrase pairs. For our corpus, we selected a random sample of true paraphrases (*label 3* or higher) from the train section. For TaPaCo, the sample covers all paraphrase candidates from the corpus, however with the restriction of taking only one, random pair from each ‘set’ of paraphrases. For Opusparcus, which is sorted by a confidence score in descending order, the sample was selected to contain the most confident 12K paraphrase candidates.⁵

In Figure 3 the length distribution of paraphrases in terms of tokens is measured for the abovementioned samples. Although the majority of paraphrases are rather short in all three corpora, we see that our corpus includes a considerably higher proportion of longer paraphrases. The average number of tokens in our corpus is 8.3 tokens per paraphrase, while it is 5.6 in TaPaCo and 3.6 in Opusparcus candidates.

In Figure 4 the paraphrase pair cosine similarity distribution is measured using TF-IDF weighted character n-grams of length 2–4. While both

⁵When we repeated the length analysis with a sample of 480K most confident pairs, the length distribution and average length remained largely unchanged, while the similarity distribution became close to flat. Without manual annotation, it is hard to tell the reason for this behavior.

TaPaCo and Opusparcus lean towards higher similarity candidates, the distribution of our corpus is more balanced including a considerably higher proportion of pairs with low lexical similarity.

6 Paraphrase Classification Baseline

In order to establish a baseline classification performance on the new dataset, we train a classifier based on the FinBERT model (Virtanen et al., 2019). Each paraphrase pair of statements A and B is encoded as the sequence [CLS] A [SEP] B [SEP], where [CLS] and [SEP] are the special marker tokens of the BERT model. Subsequently, the output embeddings of the three special tokens are concatenated together with the averaged embeddings of the tokens in A and B. These five concatenated embeddings are then propagated into four decision layers: one for the base label 2/3/4, one for the subsumption flag </>/none, and one for each the binary flag s and i. Since the flags only apply to base label 4, no gradients are applied to these layers for examples with base labels 2 and 3. We have explored also other BERT-based architectures, such as basing the classification on the [CLS] embedding only as is customary, and having a single classification layer comprising all possible base label and flag combinations. These resulted in a consistent drop in prediction accuracy, and we did not pursue them any further.

The baseline results are listed in Table 4 showing that per-class F-score ranges between 38–71%, strongly correlated with the number of examples available for each class. When interpreting the task as a pure multi-class classification, i.e. when counting all possible combinations of base label and flags as their own class, the accuracy is 54% with majority baseline being 34.3%, and the annotators’ accuracy 68.7%. The model thus positions roughly to the mid-point between the trivial majority baseline, and human performance.

7 Discussion and Future Work

In this work, we set out to build a paraphrase corpus for Finnish that would be (a) in the size category allowing deep model fine-tuning and (b) manually gathered maximizing the chance of finding more non-trivial, longer paraphrases than would be possible with the traditional automatic candidate extraction. The annotation so far took 14 person-months and resulted in little over 50,000 manually classified paraphrases. We have demon-

Label	Prec	Rec	F-score	Support
2	50.9	31.2	38.7	93
3	57.7	31.9	41.1	990
4	66.2	78.2	71.7	2149
4<	52.8	53.5	53.2	1007
4>	52.6	56.1	54.3	1136
i	51.5	36.5	42.7	329
s	51.4	28.9	37.0	249
W. avg	52.9	54.0	52.2	
Acc			54.0	

Table 4: Classification performance on the test set, when the base label and the flags are predicted separately. In the upper section, we merge the subsumption flags with the base class prediction, but leave the *i* and *s* separated. The rows *W. avg* and *Acc* on the other hand refer to performance on the complete labels, comprising all allowed combinations of base label and flags. *W. avg* is the average of P/R/F values across the classes, weighted by class support. *Acc* is the accuracy.

strated that, indeed, the corpus has longer, more lexically dissimilar paraphrases. Building such a corpus is therefore shown feasible and presently it is likely the largest manually annotated paraphrase dataset for any language, naturally at the inevitably higher data collection cost. The manual selection is only feasible for texts rich in paraphrase, and the domains and genres covered by the corpus is necessarily restricted by this condition.

In our future work, we intend to extend the manually annotated corpus, ideally roughly double its present size. We expect the pursued data size will allow us to build sufficiently accurate models, both in terms of embedding and pair classification, to gather further candidates automatically at a level of accuracy sufficient to support down-stream applications. We are also investigating further text sources, especially parallel translations outside of the present subtitle domain. The additional flags in our annotation scheme, as well as the nearly 10,000 rewrites allow for interesting further investigations in their own right.

While in the current study we concentrated on training a classifier for categorizing the paraphrases into fine-grained sub-categories, where only 2% of the paraphrases in the current release belonged to *related but not a paraphrase* category (label 2), which can be seen as a negative class

in the more traditional *paraphrase* or *not a paraphrase* classification task. In order to better account for this traditional classification task, in future work, in addition to extending the number of positive examples, we will also look into methods for expanding the training section with negative examples. While extending the data with *unrelated paraphrase* candidates (label 1) can be considered a trivial task, as more or less any random sentence pair can be considered *unrelated*, the task of expanding the data with interesting *related but not a paraphrase* candidates (label 2) is an intriguing question. One option to consider in future work is active learning, where the confidence scores provided by the initial classifier could be used to collect difficult negatives.

8 Conclusions

In this paper we presented the first entirely manually annotated paraphrase corpus for Finnish including 45,663 naturally occurring paraphrases gathered from alternative movie or TV episode subtitles and news headings. Further 7,909 hand-made rewrites are provided, turning context-dependent paraphrases into perfect paraphrases whenever possible. The total size of the released corpus is 53,572 paraphrase pairs of which 98% are manually classified to be at least paraphrases in their given context if not in all contexts.

Additionally, we evaluated the advantages and costs of manual paraphrase candidate selection from two ‘parallel’ but monolingual documents. We demonstrated the approach on alternative subtitles showing the technique being feasible for high quality candidate selection yielding sufficient amount of paraphrase candidates for the given annotation effort. We have shown the candidates to be notably longer and less lexically overlapping than what automated candidate selection permits.

The corpus is available at github.com/TurkuNLP/Turku-paraphrase-corpus under the CC-BY-SA license.

Acknowledgments

We gratefully acknowledge the support of European Language Grid which funded the annotation work. Computational resources were provided by CSC — the Finnish IT Center for Science and the research was supported by the Academy of Finland. We also thank Sampo Pyysalo for fruitful discussions and feedback throughout the project,

and Jörg Tiedemann for his generous assistance with the OpenSubtitles data.

References

- Alaa Saleh Altheneyan and Mohamed El Bachir Menai. 2019. Evaluation of state-of-the-art paraphrase identification and its application to automatic plagiarism detection. *International Journal of Pattern Recognition and Artificial Intelligence*, 34.
- Rahul Bhagat and Eduard Hovy. 2013. Squibs: What is a paraphrase? *Computational Linguistics*, 39(3):463–472.
- Mathias Creutz. 2018. Open Subtitles paraphrase corpus for six languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP 2005)*.
- Qingxiu Dong, Xiaojun Wan, and Yue Cao. 2021. ParaSCI: A large scientific paraphrase dataset for longer paraphrase generation. *arXiv preprint arXiv:2101.08382*.
- Yun He, Zhuoer Wang, Yin Zhang, Ruihong Huang, and James Caverlee. 2020. PARADE: A new dataset for paraphrase identification requiring computer science domain knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7572–7582.
- Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1224–1234.
- Nitin Madnani and Bonnie J. Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3):341–387.
- Ramtin Mehdizadeh Seraj, Maryam Siahbani, and Anoop Sarkar. 2015. Improving statistical machine translation with a multilingual paraphrase database. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1379–1390.
- Yves Scherrer. 2020. TaPaCo: A corpus of sentential paraphrases for 73 languages. In *Proceedings of The 12th Language Resources and Evaluation Conference (LREC 2020)*, pages 6868–6873.
- Sarvesh Soni and Kirk Roberts. 2019. A paraphrase generation system for EHR question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 20–29.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 2214–2218.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for Finnish. *arXiv preprint arXiv:1912.07076*.

A English Translation of Table 2

Label	Statement 1	Statement 2
4	Shockingly childish!	Astoundingly immature!
4s	I have worked for the duration of lunch.	I worked through the whole chowtime.
4i	You guys got lucky, didn't you.	Aren't you fortunate.
4>	I worked so hard for the money.	I put so much effort into work.
4<s	You rule! Come on, dude!	You are the best, Tähhä! Come on!
4is	You nicked our plant!	You stole our plants!
3	I intend to make an experiment.	I am going to test it.
2	Defeated Väyrynen vanished into the Helsinki night	Väyrynen is losing his seat in the parliament.
	Rewrites	
Orig	Can I get back to my assignments?	Can I continue?
Rew	<i>Can I get back to my assignments?</i>	<i>Can I continue working on my assignments?</i>

Table 5: English translations for annotation examples in Table 2.

Negation in Norwegian: an annotated dataset

Petter Mæhlum*, Jeremy Barnes*, Robin Kurtz†, Lilja Øvrelid* and Erik Velldal*

*University of Oslo, Department of Informatics

†National Library of Sweden, KBLab

{pettemae | jeremycb | liljao | erikve} @ifi.uio.no
robin.kurtz@kb.se

Abstract

This paper introduces NoReC_{neg} – the first annotated dataset of negation for Norwegian. Negation cues and their in-sentence scopes have been annotated across more than 11K sentences spanning more than 400 documents for a subset of the Norwegian Review Corpus (NoReC). In addition to providing in-depth discussion of the annotation guidelines, we also present a first set of benchmark results based on a graph-parsing approach.

1 Introduction

This paper introduces a new data set annotating negation for Norwegian. As shown in the example below, the annotations identify both negation *cues* (in bold) and their *scopes* (in brackets) within the sentence:

- (1) *Men kanskje **ikke** [helt troverdig] .*
But maybe not completely credible .
'But maybe not completely credible.'

The underlying corpus is the NoReC_{fine} data set (Øvrelid et al., 2020) – a subset of the Norwegian Review Corpus (NoReC) (Velldal et al., 2018) annotated for fine-grained sentiment, comprising professional reviews from a range of different domains. The new data set introduced here, named NoReC_{neg}, is the first data set of negation for Norwegian. We also present experimental results for negation resolution based on a graph-parsing approach shown to yield state-of-the-art results for other languages. All the resources described in the paper – the data set, the annotation guidelines, the models and the associated code – are made publicly available.¹

The rest of the paper is structured as follows. We start by reviewing related work on negation

for other languages in Section 2, with regards to both annotation and modeling. In Section 3 we detail our annotation guidelines, the annotation procedure and further present an analysis of inter-annotator agreement. In Section 4 we then summarize the statistics of the final annotated data set, before presenting the first benchmark results for negation resolution in Section 5. Before concluding, we finally provide a discussion of future work in Section 6.

2 Related Work

Below we discuss related work on negation, starting with datasets before moving on to modeling.

2.1 Datasets

While NoReC_{neg} is the first dataset annotated for negation for Norwegian, there are a number of existing negation datasets for a range of other languages, such as Chinese (Zou et al., 2016), Dutch (Afzal et al., 2014), English (Pyysalo et al., 2007; Vincze et al., 2008; Morante and Daelemans, 2012; Councill et al., 2010; Konstantinova et al., 2012), German (Cotik et al., 2016), Spanish (Jiménez-Zafra et al., 2018; Diaz et al., 2017), Swedish (Dalianis and Velupillai, 2010; Skeppstedt, 2011), Italian (Altuna et al., 2017), and Japanese (Matsuyoshi et al., 2014). Jiménez-Zafra et al. (2020) provide a thorough survey of existing negation datasets. A large proportion of negation corpora are based on data from the biomedical or clinical domain (Vincze et al., 2008; Dalianis and Velupillai, 2010; Cotik et al., 2016; Diaz et al., 2017). We will here focus on the corpora that are most relevant to the current annotation effort: the SFU Corpus and the ConanDoyle-neg corpus. The SFU corpus also annotates review data, hence is similar to our work in terms of text type, whereas ConanDoyle-neg is one of the most widely used datasets in the field.

The English (Konstantinova et al., 2012) and

¹https://github.com/litgoslo/norec_neg

Spanish (Jiménez-Zafra et al., 2018) parts of the SFU Review Corpus contain reviews from eight domains (books, cars, computers, cookware, hotels, movies, music, phones) which have been annotated for sentiment at document-level, as well as negation and speculation at sentence-level. The annotation scheme for negation is based primarily on the guidelines developed for the biomedical BioScope corpus (Vincze et al., 2008), which largely employ syntactic criteria for the determination of scope, choosing the maximal syntactic unit that contains the negated content. Unlike BioScope, however, negation cues are not included within the scope in SFU. The corpus does not annotate affixal cues, e.g. *im-* in *impossible*.

The English ConanDoyle-neg corpus contains Sherlock Holmes stories manually annotated for negation cues, scopes, and events (Morante and Daelemans, 2012) and was employed in the 2012 *SEM shared task on negation detection (Morante and Blanco, 2012). The annotation scheme is also based on the scheme employed for the BioScope corpus (Vincze et al., 2008), but with important modifications. In ConanDoyle-neg, the cue is not included in the scope, and it annotates a wide range of cue types, i.e., both sub-token (affixal), single token and multi-token negation cues. Scopes may furthermore be discontinuous, often an effect of the requirement to include the subject within the negation scope. This is in contrast to the annotation scheme found in the SFU corpus, where subjects are not included in the negation scope. Note that the NegPar corpus contains a re-annotated version of the ConanDoyle-neg corpus, which fixes known bugs and also adds Chinese data (Liu et al., 2018).

2.2 Modeling

Traditional approaches to the task of negation detection have typically employed a wide range of hand-crafted features, and often linguistically informed, derived from constituency parsing (Read et al., 2012; Packard et al., 2014), dependency parsing (Lapponi et al., 2012), or Minimal Recursion Semantics structures created by an HPSG parser (Packard et al., 2014). Scope resolution in particular has often been approached as a sequence labeling task, as pioneered by Morante and Daelemans (2009) and later done in several other works (Lapponi et al., 2012; White, 2012; Enger et al., 2017). More recently, neural approaches

have been successfully applied to the task. Qian et al. (2016) propose a CNN model for negation scope detection on the abstracts section of the BioScope corpus, which operates over syntactic paths between the cue and candidate tokens. Fancellu et al. (2016) present and compare two neural architectures for the task of negation scope detection on the ConanDoyle-neg corpus: a simple feed-forward network and a bidirectional LSTM. Note that these more recent neural systems disregard the task of cue detection altogether (Fancellu et al., 2016; Qian et al., 2016; Fancellu et al., 2017), relying instead on gold cues and focusing solely on the task of scope detection.

Finally, Kurtz et al. (2020) cast negation resolution as a graph parsing problem and perform full negation resolution using a dependency graph parser (Dozat and Manning, 2018) to jointly predict cues and scopes. The neural model uses a BiLSTM to create token-level representations, and then includes two feed-forward networks to create head- and dependent-specific token representations. Finally, each possible head-dependent combination is scored using a bilinear model. Despite the conceptual simplicity, this model achieves state-of-the-art results. As such, we use this model to evaluate our annotations and include further details in Section 5.

3 Annotations

In the following section we present our negation annotation effort in more detail, including the underlying source of the data. The guidelines we have developed for the annotation of negation cues and scopes in Norwegian are mainly adapted from ConanDoyle-neg (Morante and Daelemans, 2009), NegPar (Liu et al., 2018), and the Spanish SFU corpus (Jiménez-Zafra et al., 2018), modified to suit Norwegian, and with simplifications that will be discussed below. Note that while the complete set of guidelines is distributed with the corpus, we provide a brief overview below together with examples, also discussing inter-annotator agreement.

3.1 The underlying corpus

The negation annotations described below are added to the existing NoReC_{fine} data set² (Øvreliid et al., 2020) – a subset of the Norwegian Review Corpus (NoReC) annotated for fine-grained sentiment. The negation layer of the corpus is named

²https://github.com/lrgoslo/norec_fine

NoReC_{neg}. The full NoReC corpus (Velldal et al., 2018) contains professional reviews from several Norwegian online news sites, spanning a range of different domains, like music, literature, products, movies, restaurants, and more. While NoReC contains more than 43,000 full-text reviews, the subset annotated in NoReC_{fine}, and hence also NoReC_{neg}, includes 414 full reviews, comprising 11,346 sentences. Note that there are two official standards for written Norwegian; Bokmål (the majority variant) and Nynorsk. While the data set contains a majority of documents written according to the Bokmål standard, four Nynorsk documents are also included.

3.2 Negation in Norwegian

Since our starting point for guideline development is English, we will here discuss linguistic differences between the expression of negation in the two languages. Generally speaking, Norwegian negation does not differ greatly from English. The main means of negating a proposition is by using adverbs, prepositions and quantifiers. The largest differences between the two are syntactic in nature and concern the placement of adverbials, caused by the fact that Norwegian, unlike English, is a V2-language. One clear difference with practical consequences is that certain Norwegian negation cues inflect for grammatical gender and number, notable examples being *ingen* (*ingen, inga, intet*) ‘no’ and *løs* (*-løs, -løst, -løse*) ‘-less’, as seen in example (2) for the affixally negated (a) *meningsløst* ‘meaningless’ with the neuter ending, (b) *hensynsløse* ‘inconsiderate’ with plural inflection, and (c) *smakløs* ‘tasteless’ with no inflection. This property of Norwegian means that there are likely a larger number of different tokens functioning as cues in Norwegian, as compared to English.

- (2) (a) [...] blir ganske menings**løst**
 (b) [...] hensynsl**øse** regnskog-ødeleggere
 (c) [...] men ikke smak**løs** .

The discussion of negation in the Norwegian Reference Grammar (Faarlund et al., 1997) is largely limited to a selected few of the possible cues, e.g., *ikke* ‘not’, *ingen* ‘none, no-one’ and related forms, and the preposition *uten* ‘without’. Golden et al. (2014) contains a brief comment on lexical negation, where they mention *nektende verb* ‘negating verbs’. They also mention negative polarity items under a discussion of separate words and expressions in negations.

3.3 Negation cues

A negation cue is a word or a set of words that serve to signal negation. In our annotation scheme we annotate both single token cues, such as adverbs like *ikke* ‘not’, *aldri* ‘never’, prepositions, e.g., *uten* ‘without’, and quantifiers like *ingen* ‘no’. We also annotate multi-word cues, such as (*på*) *ingen måte*, ‘in no way’, as well as morphological or affixal negation cues, i.e. affixes such as *u-* ‘un-/dis-/non-’ and *-løs* ‘-less’. Example (3) shows the widely used negative adverb *aldri* ‘never’, which scopes over the whole sentence, including the subject *Jeg* ‘I’, whereas (4) exemplifies the negative determiner *ingen* ‘no’ which occurs in two conjoined noun phrase objects, where both negation cues scope over the following noun as well as the preceding subject and main verb.

- (3) [*Jeg har*] **aldri** [*hørt henne synge bedre*
 I have never heard her sing better
fra en scene]
 from a stage
 ‘I have never heard her sing better from a stage’
- (4) [*Den stiller*] **ingen** [*spørsmål*] og [*gir*]
 It asks no questions and gives
ingen [*svar*] .
 no answers.
 ‘It asks no questions and gives no answers.’

Multi-word cues Multi-word cues are negation cues that span more than one token. These may further be *discontinuous*, as in the case of (*h*)*verken ... eller* ‘neither ... nor’, as seen in example (5). As noted by Morante and Daelemans (2012), multi-word cues tend to be fixed/idiomatic expressions – an observation that is largely true for Norwegian as well. One practical difference between the annotation scheme in Morante and Daelemans (2009) and ours, is that we omit prepositions and particles related to these expressions, as in (6), in favor of creating less variation that might create noise in the data, especially in cases where multiple prepositions are associated with similar cues and the association is less fixed.

- (5) [...] **verken** [*manus*] **eller** [*skuespillere*
 [...] neither script nor actors
trekker oss inn på en engasjerende
 pull us in on a engaging
måte] .
 method .
 ‘[...] neither script nor actors pull us inside in an engaging way’

- (6) *Og mest av alt fraværet av [mer
And most of all the absence of more
enn bare et kvarter musikk] .
than just a quarter music .
'And most of all, the absence of more than just
15 minutes of music.'*

Affixal cues We annotate both free-standing and affixal negation cues. The affixal cues form a rather closed group of cues, with the prefix *u-* and the suffix *-løs* being the most common. However, our annotations show that there is lexical variation, with less common cues such as *-fri* ‘-free’ and *-tom* ‘-empty’.

Negation vs. Modality One difficulty in annotating cues is to separate between cases of negation in isolation and cases where negation and modality interact. Cases where modality and negation are inseparable, as in *neppe* ‘barely’ are not annotated, but cases of negation where the modality can be separated, either by it scoping over the negation, or the negation scoping over it, were annotated as negations.

Lexical negation As mentioned above, the discussion of lexical negation in a Norwegian context is limited. We borrow the term ‘lexical negation’ from Jiménez-Zafra et al. (2020), who split cues into syntactic, lexical, and morphological/affixal, and use the lexical category to mean words that fall outside the ‘syntactic’ and more frequent cues, like negative adverbs and determiners. Examples from Norwegian include verbal constructs, e.g., *la være* ‘refrain from’ or *forsvinne* ‘disappear’ as in (7), and nouns such as *mangel* ‘lack’.

- (7) ...[Irritasjonen] *forsvant* da maten
...the.irritation disappeared when the.food
kom .
arrived .
'...The irritation disappeared when the food
arrived.'

Lexicalization and idioms The words that are used as negation cues might also have other functions, and are in some cases part of fossilized expressions. The annotators were instructed to refrain from annotating affixal cues that no longer signal negation. Lexicalization, in particular, is a challenge when it comes to affixal negation, as it can be difficult even for native speakers to judge whether something should be treated as a negation or not. Some cases are clearer than others, such as *uansett* ‘regardless’, which stems from

ansett ‘viewed/respected’, which it clearly does not negate, on the one hand, and on the other hand *usikker* ‘uncertain’, whose non-negated form *sikker* ‘certain’ is also frequent. The absence of the non-negated version of the lemma in the language might be a good indicator of lexicalization, and annotators were instructed to avoid annotating such words.

In addition to lexicalized items, there are also cases where a cue can have more than one meaning. One frequent case is the prefix *u-* with nominal roots, a construction that usually results in nouns meaning bad *x*, as in *uår* lit. ‘un-year’, which means ‘a bad year’, or *uvenn*, lit. ‘un-friend’, meaning ‘enemy’. The annotators were instructed to try and dismantle the word in order to see if the word made sense without the negative prefix, in which case it would indicate that it is not completely lexicalized. Even so, these are often difficult judgements for the annotators to make. Furthermore, nominalizations of negated adjectives, such as *uttrykksløshet* ‘expressionlessness’ and *umenneskelighet* ‘inhumanity’ were not to be annotated.

Table 1 presents the ten most common cues found in the corpus, where we find both affixal and single token cues. We see that variation in the data is further caused by spelling differences. The adverb *ikke* ‘not’ can also be used affixally, often, but not always, with a hyphen, as in *ikke-produksjonsklart* ‘not-production-ready’. The variation is also due in part to the two language varieties present in the dataset, e.g. in the case of Bokmål *ikke* ‘not’ and Nynorsk *ikkje* ‘not’.

3.4 Negation scopes

The scope of a negation is the part of a sentence that has its truth value inverted by the presence of a negation cue. In our annotation scheme, cues are never part of the scope. Subjects are included in the scope if the negation scopes over the main verb, which usually means that the whole proposition is negated, and if the subject or object of a sentence is negated by a determiner or similar, the whole sentence is in the scope, apart from certain fixed elements discussed below. Phrase linking conjunctions are not included. Furthermore, scopes tend to be discontinuous. In many cases this is simply due to the fact that in most sentences, the subject precedes the negation cue, while the predicate follows it.

Cue	Trans.	Frequency	Amb. Rate
ikke	not	1,364	3
u-	un-/dis-/non-	514	83
uten	without	190	0
ingen	none/nobody	134	0
-løs	-less	123	5
aldri	never	95	6
mangle	lack	43	14
ingenting	nothing	23	0
ikkje	not	23	0
verken	neither	21	30

Table 1: List of the 10 most common cues found in the corpus, their translation to English, their frequency as a cue, as well as their ambiguity rate (Amb. Rate), which is defined as $1 - (\text{the frequency as a cue} / \text{the absolute frequency}) \times 100$.

Implicit scope The scope of a cue can be implicit, meaning it is understood from the context. In practice the scope is often expressed in a sentence before or after the cue itself. This is in particular the case with the interjection *nei* ‘no’, which usually refers back to the proposition it negates. Since our annotation does not span across sentence boundaries, the scope is annotated as implicit in these types of cases.

Subordinate clauses If the negation cue modifies a verb in a subordinate clause, the whole subordinate clause, except the initial subjunction, is part of the scope, see (8) below.

- (8) *Det føles derfor som et pluss at*
 It feels therefore like a plus that
[plata] ikke [er særlig lang] .
 the.record not is especially long .
 ‘It therefore feels like a bonus that the record is not especially long.’

Modifying subjects and objects If a cue, typically a determiner, modifies the subject or the object of a sentence, the whole clause that contains that subject or object is part of the scope, as in (9) below. Note that certain elements, such as subjunctions, conjunctions and sentence adverbs might still not be included.

- (9) *[Her viser Selbekk] ingen [nåde] .*
 Here shows Selbekk no mercy .
 ‘Here, Selbekk shows no mercy.’

Cue as subject or object In cases where the subject or object are also negation cues, the cue is not included in the scope, see (10).

- (10) *Og ingen [er hardere enn Regan] .*
 And nobody is tougher than Reagan .
 ‘And nobody is tougher than Reagan.’

Exception items The annotation of exception items, such as *untatt* ‘except’ and *bortsett (fra)* ‘except (for)’ depends on whether they are within the scope of a negation cue or not. When the item is not within the scope of another cue, it incurs a negation, as in (11). This closely follows the annotation found in Morante and Daelemans (2012) and Liu et al. (2018).

- (11) *Sportsseter - som gir god støtte*
 Sports-seats - which give good support
unntatt [lårstøtten for høyvokste
 except the.thigh-support for high-grown
personer] .
 people
 ‘Sport seats - which give good support, except for the thigh support for tall people’

When exception items are found within the scope of another negation cue, however, they remove the elements they scope over from the scope of the other negation.

Sentential adverbs and adverbs scoping over negation Two types of adverbs pose certain challenges: sentential adverbs and adverbs that indicate modality. Sentential adverbs such as *heldigvis* ‘fortunately’ as in (12) are not part of the propositional value of a sentence, but rather function to comment on it (Faarlund et al., 1997). Therefore they are usually outside the scope of the negation, as is shown by (12):

- (12) *Heldigvis [skjer dette] nesten aldri .*
 Fortunately happens this almost never .
 ‘Fortunately, this almost never happens.’

Modal adverbs such as *kanskje* ‘maybe’ can occur both within and outside of the scope of a negation cue, and in these cases the annotators were asked to paraphrase in order to pinpoint the placement of these adverbs.

Negation raising Negation raising is the phenomenon where a negator is “raised” further up in a syntactic tree, which in the case of Norwegian means further towards the beginning of a sentence. What characterizes these types of constructions is

that the negation is adjacent to the verb in the main sentence, even though the negation only scopes over a subsequent subordinate clause. This happens frequently in Norwegian, as in English, with mental state verbs like *mene* ‘think’, *tro* ‘believe’, as in (13).

- (13) *Harry Hole tror imidlertid ikke at Harry Hole believes however not that [saken kan være så enkel] [...] the case can be so simple [...] ‘Harry Hole, however, does not believe that the case is that simple!’*

Expletive subjects In Norwegian, as in other Scandinavian languages, there are several types of linguistic constructions that involve an expletive subject. A commonly used mechanism in these languages is extraposition, where a clausal argument is postposed, and a formal, semantically void subject *det* ‘it’ or *der* ‘there’ functions as the syntactic subject, as in (14). Here we do not treat the expletive subject as the subject of the negated proposition, instead only the extraposed subordinate clause is in scope of the negation. Since *det* ‘it’ is ambiguous in the sense that it can, in fact, also be referential, the annotators have to assess referentiality during annotation.

- (14) *Det [er] aldri [kjedelig å se gode It is never boring to see good replikker fremført i vakre lines performed in beautiful omgivelser] . ‘It is never boring to see good lines performed in beautiful surroundings.’*

Negation in conditional, interrogative, and imperative sentences In the annotation scheme of Morante and Daelemans (2012), they do not annotate negation in non-factual sentences, i.e., conditional, interrogative and imperative sentences. We have chosen to include all negation regardless of its factuality. We believe that negation has implications beyond asserting the factuality of a proposition, and it can be useful for sentiment analysis, among other tasks. For instance, in example (15), the negation is under the scope of the conditional *hvis* ‘if’, but is still marked, even though it is not a factual proposition.

- (15) *Hvis [folk] ikke [hadde snakket til if people not had talked to hverandre i det hele tatt] [...] each.other in the whole taken [...] ‘If people had not talked to each other at all [...]’*

Negative polarity items (NPIs) NPIs are lexical entities that are used together with negation cues, and which usually render the sentence ungrammatical should the negation cues be removed without further change. In our annotation scheme, they are contained within the scope of the negation cue. In Norwegian, the negative adverb *ikke* ‘not’ in combination with the determiner *noe/noen* ‘some/any’ is a common negative polarity item. However, the most common type of NPIs are adverbs such as *i det hele tatt* ‘at all’, as in (16), that serve to strengthen the negation.

- (16) *[Han kan] ikke [synge i det hele He can not sing in the whole tatt] . taken . ‘He cannot sing at all.’*

Foreign language citations The annotated texts frequently contain titles of various products, such as ‘Never Run Away’. These cases of foreign language negation cues are not annotated.

Negation cues not indicating negation It is not uncommon for negation cues to be part of expressions that do not indicate negation in combination, e.g., certain fixed expressions such as *hvis ikke* ‘otherwise’. Other borderline cases such as the focus marker *ikke bare* ‘not only’ and the expression *ingen tvil* ‘no doubt’, were included after discussion, as they are analyzed as introducing a negated reading.

Affixal scope The scope of affixal items is annotated in a slightly different way compared to other cues. If an affixally negated adjective is the predicate, then the whole sentence is included within its scope. If it is part of a noun phrase, then only that noun phrase is inside the scope. Additional adjectives or adverbs in the sentence fall outside the scope, as in (17).

- (17) *Passasjerene er for oss u[kjente] , the.passengers are for us unknown , anonyme [fjes] . anonymous faces . ‘The passengers are unknown faces to us.’*

3.5 Annotation Procedure

The annotation was performed by several hired student research assistants with a background in linguistics and with Norwegian as their native language. All 414 documents in the original dataset,

comprising 11,346 sentences, were annotated independently by two annotators in parallel. The doubly annotated documents were then adjudicated by a third annotator after a final round of discussions concerning difficult cases. Annotators had the possibility to discuss any potential problems during both the annotation and adjudication period, but were encouraged to follow the guidelines as strictly as possible. The annotation and adjudication were both performed using the web-based annotation tool Brat (Stenetorp et al., 2012).

3.6 Inter-annotator agreement

We have measured the inter-annotator agreement over the full (doubly annotated) dataset in terms of both F_1 and κ scores for cues, full scopes, and scope tokens. The scores show that annotators agree to a very high degree on the identification of cues (0.995 F_1 , 0.841 κ). When it comes to negation scopes, the agreement is lower when measured towards full and exact spans (0.632 F_1 , 0.34 κ), but quite high when measured on the token-level (0.912 F_1 , 0.803 κ).

Due to the adjudication phase of the annotation process, we also have insight into the sources of disagreements between the annotators. As noted above, agreement between annotators is generally high when it comes to cue detection, but surprising disagreements can be seen. These are most likely due to the guidelines being improved as the annotations continued to uncover new challenges. There seems to be a clear tendency for annotators to disagree on less common cues, such as verbs and nouns that indicate negation, as opposed to the more often discussed adverbs and determiners. The annotators rarely agreed on less frequent lexical items such as *forsvinne* ‘disappear’ and *takke nei til* ‘say no to’. However, the disagreements also reflect discussions concerning the inclusion or omission of prepositions, in addition to cue span errors. Annotators generally agree on the more frequent cues. The prefix *u-* ‘un-/dis-/non-’, seems to have a disproportionately large disagreement score, but discussions among the annotators indicate that this is likely due to prefixes being more difficult to detect when annotating than isolated whole-word tokens. Disagreement is also found regarding modal elements, such as *knapt* ‘barely’ (almost not) and *for...til* ‘too...to’ (cannot be).

4 Corpus statistics

Table 2 summarizes the statistics for the final annotated data set. Of the 11,346 sentences in the corpus, we see that just above 20% of them are negated. Out of the negated sentences, 13% contain multiple instances of negation. While, as expected, the number of tokens in a *cue* averages to 1, the average length of *scopes* is close to 7 (with a maximum observed length of 53). Note, however, that a small number of cues (1.4%) also have empty (‘null’) scopes. We report both any kind of discontinuous scopes (disc.) and true discontinuous scopes (true disc.), where the latter does not count scopes which are only discontinuous because of an intervening cue. While discontinuous scopes are very frequent (70% of scopes), truly discontinuous scopes are much fewer (21%). We see that affixal negation is quite widespread in NoReC_{neg}, comprising almost 25% of the cues. Moreover, just above 11% are multi-word cues. While most cues are not particularly ambiguous, e.g., *ikke* ‘not’, *uten* ‘without’, others, such as *u-* ‘un-/dis-’, *mangle* ‘lack’ or *verken* ‘neither’ can have rather high rates of ambiguity (meaning that they can occur with both negated and non-negated readings).

5 Experiments

5.1 Modeling approach

In order to benchmark the dataset, we use the semantic graph parsing approach to negation detection proposed by Kurtz et al. (2020), see Section 2. Besides the baseline graph representation originally proposed (*point-to-root*), where all elements of the scope have arcs that point to the cue, we propose several variants. For *head-first*, we set the first token of the cue as the root node, and similarly set the first token in the scope as the head of the span. All elements within the span have arcs that point to the head, and heads have arcs that point to the root. *head-final* is similar, but instead sets the final tokens of spans as the heads. There can be several roots per sequence and not all tokens are connected. Finally, we enrich the dependency labels to distinguish edges that are internal to a holder/target/expression span from those that are external and perform experiments by adding an ‘in label’ to non-head nodes within the graph, which we call *+inlabel*.

	Sentences		Cues						Scopes					
	#	neg.	#	avg.	max	disc.	mult.	affixal	#	avg.	max	disc.	true disc.	null
train	8,543	1,768	2,025	1	3	19	228	508	1,995	6.9	44	1,403	423	30
dev	1,531	301	342	1	2	0	39	88	339	7.1	53	236	85	3
test	1,272	263	305	1	2	2	37	69	301	6.5	27	203	58	4
total	11,346	2,332	2,672	1	3	21	304	665	2,635	6.9	53	1,842	566	37

Table 2: Statistics of the dataset – per split and in total – including total number of sentences (#), number of sentences that contain negation (neg.), as well as the number (#) of cues and scopes, along with their average and maximum lengths in tokens. Additionally, we include the number of discontinuous cues and scopes (disc.) as well as true discontinuous (true disc.) for scopes which we discuss in Section 4. Finally, we detail the number of sentences that have multiple cues (mult.), the number of affixal cues, and the number of cues that have no scope (null).

5.2 Results

The negation parser is evaluated using the metrics from the *SEM 2012 shared task (Morante and Blanco, 2012): cue-level F_1 (CUE), scope token F_1 over individual tokens (ST), and the full negation F_1 (FN) metric. In contrast to the *SEM 2012 shared task we do not annotate negated events, meaning that FN only requires an exact match of the negation’s cue(s) and, if present, all its scope tokens. We run each experiment five times with different random seeds and report an averaged F_1 score and its standard deviation in Table 3.

The simplest graph representation *point-to-root* generally performs best, most visibly in FN F_1 (66.8). We attribute the variation in performance to a loss of information in the *head-first* and *head-final* variants, making it impossible to retrieve the correct governing negation cue for partially overlapping scopes, thus lowering the score.

In order to see whether these performance differences are statistically significant, we perform bootstrap significance testing (Berg-Kirkpatrick et al., 2012) resampling the test set 10^6 times while setting the significance threshold to $p = 0.05$. Comparing *point-to-root* to *head-first* and *head-final* shows that while the differences seem substantial they are not statistically significant.

A manual error analysis on *point-to-root* shows that the model tends not to predict infrequent cues, e.g., *null* ‘zero’, *istedenfor* ‘instead-of’, *savnet* ‘missing’, while it overpredicts frequent cues, e.g., *ikke* ‘not’, *ingen* ‘no’, as well as overgeneralizing the affixal negation *u-* ‘un-/dis-/non-’ to other words that begin with ‘u’, but are not negated, e.g., *utfrika* ‘freaked-out’, *unnagjort* ‘finished’. The model also tends to predict slightly shorter scopes (an average of 6.5 tokens for predicted scopes ver-

	CUE	ST	FN
<i>point-to-root</i>	93.4 (0.5)	83.6 (0.7)	66.8 (0.8)
<i>head-first</i>	92.7 (0.3)	81.9 (1.4)	65.5 (0.6)
<i>+inlabel</i>	92.7 (0.7)	81.8 (1.0)	65.0 (2.2)
<i>head-final</i>	92.7 (0.6)	82.7 (1.8)	64.8 (3.1)
<i>+inlabel</i>	93.1 (0.3)	82.2 (1.5)	65.8 (0.8)

Table 3: Results of our negation parser using the various graph representations. The results are averaged over 5 runs, additionally reporting standard deviation.

sus 6.7 for gold scopes), while the most common scope-related errors derive from discontinuous scopes, where the model fails on 75.4%. These errors are often due to inversions with the expletive ‘det’, which is not considered in scope. Although rare (4 examples in test), multi-word cues are also challenging, and the graph model only correctly predicted one of the four. Finally, affixal cues can pose a challenge as well, with the model failing on 67.1% of the sentences containing affixal negation.

6 Future work

As mentioned previously, the underlying corpus NoReC_{fine} is annotated for fine-grained sentiment, including opinion holders, targets, sentiment expressions, and positive/negative polarity. The fact that negation is among the most important compositional phenomena that can affect sentiment in terms of shifting polarity values motivated the choice of this particular dataset for adding the negation annotations. In future work we plan to

further investigate the co-dependencies between negation and sentiment, both through analyzing the existing annotations and through joint modeling.

7 Summary

This paper has introduced the first annotated dataset of negation for Norwegian, NoReC_{neg}, where negation cues and their corresponding in-sentence scopes have been annotated across more than 11K sentences spanning more than 400 documents; a subset of the Norwegian Review Corpus (NoReC). In addition to providing in-depth discussion of the annotation guidelines, we have also presented a first set of benchmark results based on a graph-parsing approach.

Acknowledgements

This work has been carried out as part of the SANT project (Sentiment Analysis for Norwegian Text), funded by the Research Council of Norway (grant number 270908). We also want to express our gratitude to the annotators: Anders Næss Evensen, Helen Ørn Gjerdrum, Petter Mæhlum, Lilja Charlotte Storset, Carina Thanh-Tam Truong, and Alexandra Wittemann.

References

- Zubair Afzal, Ewoud Pons, Ning Kang, Miriam CJM Sturkenboom, Martijn J Schuemie, and Jan A Kors. 2014. ContextD: an algorithm to identify contextual properties of medical terms in a Dutch clinical corpus. *BMC bioinformatics*, 15(1):1–12.
- Begoña Altuna, Anne-Lyse Minard, and Manuela Speranza. 2017. The scope and focus of negation: A complete annotation framework for Italian. In *Proceedings of the Workshop Computational Semantics Beyond Events and Roles*, pages 34–42, Valencia, Spain.
- Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An Empirical Investigation of Statistical Significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea.
- Viviana Cotik, Roland Roller, Feiyu Xu, Hans Uszkoreit, Klemens Budde, and Danilo Schmidt. 2016. Negation detection in clinical reports written in german. In *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM2016)*, pages 115–124.
- Isaac Council, Ryan McDonald, and Leonid Velikovich. 2010. What’s great and what’s not: learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pages 51–59, Uppsala, Sweden.
- Hercules Dalianis and Sumithra Velupillai. 2010. How Certain are Clinical Assessments? Annotating Swedish Clinical Text for (Un)certainties, Speculations and Negations. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*.
- Noa P Cruz Diaz, Roser Morante, Manuel J Mana López, Jacinto Mata Vázquez, and Carlos L Parra Calderón. 2017. Annotating negation in Spanish clinical texts. In *Proceedings of the workshop computational semantics beyond events and roles*, pages 53–58, Valencia, Spain.
- Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia.
- Martine Enger, Erik Velldal, and Lilja Øvrelid. 2017. An open-source tool for negation detection: a maximum-margin approach. In *Proceedings of the EACL workshop on Computational Semantics Beyond Events and Roles (SemBEaR)*, pages 64–69, Valencia, Spain.
- Jan Terje Faarlund, Svein Lie, and Kjell Ivar Vannebo. 1997. *Norsk referansegrammatikk*. Universitetsforlaget, Oslo, Norway.
- Federico Fancellu, Adam Lopez, and Bonnie Webber. 2016. Neural networks for negation scope detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 495–504, Berlin, Germany.
- Federico Fancellu, Adam Lopez, Bonnie Webber, and Hangfeng He. 2017. Detecting negation scope is easy, except when it isn’t. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 58–63, Valencia, Spain.
- Anne Golden, Kirsti Mac Donald, and Else Ryen. 2014. *Norsk som fremmedspråk: Grammatikk*, 4 edition. Universitetsforlaget, Oslo, Norway.
- Salud María Jiménez-Zafra, Roser Morante, María Teresa Martín-Valdivia, and L Alfonso Ureña-López. 2020. Corpora annotated with negation: An overview. *Computational Linguistics*, 46(1):1–52.
- Salud María Jiménez-Zafra, Mariona Taulé, M Teresa Martín-Valdivia, L Alfonso Ureña-López, and M Antónia Martí. 2018. SFU Review SP-NEG: a Spanish corpus annotated with negation for sentiment analysis. a typology of negation patterns. *Language Resources and Evaluation*, 52(2):533–569.

- Natalia Konstantinova, Sheila C.M. de Sousa, Noa P. Cruz, Manuel J. Maña, Maite Taboada, and Ruslan Mitkov. 2012. A review corpus annotated for negation, speculation and their scope. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 3190–3195, Istanbul, Turkey.
- Robin Kurtz, Stephan Oepen, and Marco Kuhlmann. 2020. End-to-end negation resolution as graph parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 14–24, Online.
- Emanuele Lapponi, Erik Velldal, Lilja Øvrelid, and Jonathon Read. 2012. UiO2: Sequence-labeling negation using dependency features. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, pages 319–327, Montreal, Canada.
- Qianchu Liu, Federico Fancellu, and Bonnie Webber. 2018. NegPar: A parallel corpus annotated for negation. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*, pages 3464–3472, Miyazaki, Japan.
- Suguru Matsuyoshi, Ryo Otsuki, and Fumiyo Fukumoto. 2014. Annotating the Focus of Negation in Japanese Text. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 1743–1750, Reykjavik, Iceland.
- Roser Morante and Eduardo Blanco. 2012. *SEM 2012 shared task: Resolving the scope and focus of negation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 265–274, Montréal, Canada.
- Roser Morante and Walter Daelemans. 2009. A meta-learning approach to processing the scope of negation. In *Proceedings of the 13th Conference on Computational Natural Language Learning*, Boulder, USA.
- Roser Morante and Walter Daelemans. 2012. ConanDoyle-neg: Annotation of negation cues and their scope in Conan Doyle stories. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, Istanbul, Turkey.
- Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2020. A fine-grained sentiment dataset for Norwegian. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5025–5033, Marseille, France.
- Woodley Packard, Emily M. Bender, Jonathon Read, Stephan Oepen, and Rebecca Drīdan. 2014. Simple negation scope resolution through deep parsing: A semantic solution to a semantic problem. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, USA.
- Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. 2007. Bioinfer: a corpus for information extraction in the biomedical domain. *BMC bioinformatics*, 8(1):1–24.
- Z. Qian, P. Li, Q. Zhu, G. Zhou, Z. Luo, and W. Luo. 2016. Speculation and negation scope detection via convolutional neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, USA.
- Jonathon Read, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2012. UiO1: Constituent-based discriminative ranking for negation resolution. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, Montreal, Canada.
- Maria Skeppstedt. 2011. Negation detection in Swedish clinical text: An adaption of NegEx to Swedish. *Journal of Biomedical Semantics*, 2 Suppl 3.
- Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: A Web-based Tool for NLP-assisted Text Annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France.
- Erik Velldal, Lilja Øvrelid, Eivind Alexander Bergem, Cathrine Stadsnes, Samia Touileb, and Fredrik Jørgensen. 2018. NoReC: The Norwegian Review Corpus. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference*, pages 4186–4191, Miyazaki, Japan.
- V. Vincze, G. Szarvas, R. Farkas, G. Móra, and J. Csirik. 2008. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC bioinformatics*, Suppl 11.
- James Paul White. 2012. UWashington: Negation resolution using machine learning methods. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, Montreal, Canada.
- Bowei Zou, Guodong Zhou, and Qiaoming Zhu. 2016. Research on chinese negation and speculation: corpus annotation and identification. *Frontiers of Computer Science*, 10(6):1039–1051.

Short Papers

What Taggers Fail to Learn, Parsers Need the Most

Mark Anderson Carlos Gómez-Rodríguez

Universidade da Coruña, CITIC

FASTPARSE Lab, LyS Research Group,

Departamento de Ciencias de la Computación y Tecnologías de la Información

{m.anderson, carlos.gomez}@udc.es

Abstract

We present an error analysis of neural UPOS taggers to evaluate why using gold standard tags has such a large positive contribution to parsing performance while using predicted UPOS tags either harms performance or offers a negligible improvement. We evaluate what neural dependency parsers implicitly learn about word types and how this relates to the errors taggers make to explain the minimal impact using predicted tags has on parsers. We also present a short analysis on what contexts result in reductions in tagging performance. We then mask UPOS tags based on errors made by taggers to tease away the contribution of UPOS tags which taggers succeed and fail to classify correctly and the impact of tagging errors.

1 Introduction

Part-of-speech (POS) tags have commonly been used as input features for dependency parsers. They were especially useful for non-neural implementations (Voutilainen, 1998; Dalrymple, 2006; Alford and Béchet, 2012). However, the efficacy of POS tags for neural network dependency parsers is less apparent especially when utilising character embeddings (Ballesteros et al., 2015; de Lhoneux et al., 2017). Universal POS (UPOS) tags have still been seen to improve parsing performance but only if the predicted tags come from a sufficiently accurate tagger (Dozat et al., 2017).

Typically using predicted POS tags has offered a nominal increase in performance or has had no impact at all. Smith et al. (2018) undertook a thorough systematic analysis of the interplay of UPOS tags, character embeddings, and pre-trained word embeddings for multi-lingual Universal Dependency (UD) parsing and found that tags offer

a marginal improvement for their transition based parser. However, Zhang et al. (2020) found that the only way to leverage POS tags (both coarse and fine-grained) for English and Chinese dependency parsing was to utilise them as an auxiliary task in a multi-task framework. Further, Anderson and Gómez-Rodríguez (2020) investigated the impact UPOS tagging accuracy has on graph-based and transition-based parsers and found that a prohibitively high tagging accuracy was needed to utilise predicted UPOS tags. Here we investigate whether dependency parsers inherently learn similar word type information to taggers, and therefore can only benefit from the hard to predict tags that taggers fail to capture. We also investigate what makes them hard to predict.

2 Methodology

We performed two experiments. The first was an attempt to compare what biaffine parsers learn about UPOS tags by fine-tuning them with tagging information and comparing their errors with those from normally trained UPOS taggers. The second experiment attempted to evaluate the impact tagging errors have by either masking errors or using the gold standard tags for erroneously predicted tags while masking all other tags.

Data We took a subset of UD v2.6 treebanks consisting of 11 languages, all of which are from different language families (Zeman et al., 2020): Arabic PADT (ar), Basque BDT (eu), Finnish TDT (fi), Indonesian GSD (id), Irish IDT (ga), Japanese GSD (ja), Korean Kaist (ko), Tamil TTB (ta), Turkish IMST (tr), Vietnamese VTB (vi), and Wolof WTB (wo). We used pre-trained word embeddings from fastText (for Wolof we had to use the previous Wiki version) (Bojanowski et al., 2017; Grave et al., 2018). We compressed the word embeddings to 100 dimensions with PCA.

	Tagger	Tagger-FT	Parser
Arabic	96.71	96.52	93.73
Basque	95.35	95.18	88.09
Finnish	96.92	96.62	92.24
Indonesian	93.72	93.79	91.98
Irish	92.84	92.80	88.24
Japanese	97.94	97.85	92.80
Korean	95.09	94.26	86.93
Tamil	89.29	87.28	75.41
Turkey	95.10	94.98	86.14
Vietnamese	87.85	87.63	83.40
Wolof	93.85	93.79	85.81

Table 1: Tagging accuracies for tagger trained normally (Tagger), “fine-tuning” a newly initialised MLP for the trained taggers (Tagger-FT), and for parsers fine-tuned to predict tags (Parser).

Experiment 1: Error crossover We trained parsers and taggers on the subset of UD treebanks described above. We then took the parser network and replaced the biaffine structure with a multi-layer perceptron (MLP) to predict UPOS tags. We froze the network except for the MLP and fine-tuned the MLP with one epoch of learning, which is similar to the process used in Vania et al. (2019). We train for only one epoch to balance training the MLP to decode what the system already has encoded without giving it the opportunity to encode more information. We repeated this for the tagger networks (replacing their MLP with a randomly initialised MLP) to validate this fine-tuning procedure. We then compared the tagging errors of both the parsers fine-tuned for tagging and the original taggers. We also undertook an analysis of the errors from the normal taggers which included looking at the impact out-of-vocabulary, POS tag context, and a narrow syntactic context. We define the contexts in Section 3.

Experiment 2: Masked tags We then used the output from the taggers from Experiment 1 to train different parsers. We trained parsers using all the predicted tags, using only the gold standard tags the taggers failed to predict (for both the standard taggers and parsers fine-tuned for tagging), using predicted tags from the standard taggers but masking the errors, and training with all gold standard tags. Note that the respective sets of POS tags were used at both training and inference time. We also trained parsers with no tags as a baseline.

Network details Both the taggers and parsers use pre-trained word embeddings and randomly-initialised character embeddings. The parsers use

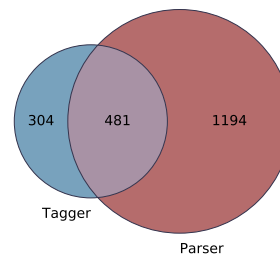


Figure 1: Average union of tagging errors for parser fine-tuned for tagging and fully-trained tagger (standard deviation: 159 for tagger error, 715 for parser, and 242 for union).

UPOS tag embeddings as specified in the experimental details. The character and tag embeddings are randomly initialised. The parsers consist of the embedding layer followed by BiLSTM layers and then a biaffine mechanism (Dozat and Manning, 2017). The taggers are similar but with an MLP following the BiLSTMs instead. We ran a small hyperparameter search using *fi*, *ga*, *tr*, and *wo* and using their respective development data. This resulted in 3 BiLSTM layers with 200 nodes, 100 dimensions for each embedding type with 100 dimension input to the character LSTM. The arc MLP of the biaffine structure had 100 dimensions, 50 for the relation MLP. Dropout was 0.33 for all layers. Learning rate was 2×10^{-3} , β_1 and β_2 were 0.9, batch size was 30, and we trained both taggers and parsers for 200 epochs but with early stopping if no improvement was seen after 20 epochs. Models were selected based on the performance on the development set.

3 Results and discussion

Experiment 1: Error crossover Table 1 shows the tagging performance for the normally trained taggers, the re-fine-tuned taggers, and the fine-tuned parser taggers. The re-fine-tuned taggers achieve relatively similar performance to the original taggers, which suggests that this procedure does allow us to develop a decoder that captures

	All	Open	Closed	Other
Tagger	8,637	6,434	1,867	336
Parser	18,426	15,181	2,816	429
Total	171,373	101,965	46,362	23,046

Table 2: Error (Parser, Tagger) and total (Total) counts across all data per word class of gold tag.

		Error Types				Errors	Tokens
ar	noun→x 197	x→noun 139	noun→adj 108	adj→x 78	adj→noun 60	931	28.3K
eu	propn→noun 145	verb→aux 113	noun→adj 101	aux→verb 100	adj→noun 94	1134	24.4K
fi	propn→noun 56	noun→propn 53	noun→adj 43	adj→noun 39	noun→verb 37	649	21.1K
id	propn→noun 147	noun→propn 92	adj→noun 47	noun→adj 34	verb→noun 23	740	11.8K
ga	propn→noun 184	noun→propn 53	noun→adj 53	adj→noun 38	noun→pron 36	724	10.1K
ja	noun→adv 52	propn→noun 24	noun→adj 22	adj→noun 22	aux→verb 20	269	13.0K
ko	noun→propn 252	propn→noun 145	verb→adj 133	aux→verb 78	cconj→sconj 75	1394	28.4K
ta	noun→propn 24	aux→verb 22	propn→noun 17	noun→verb 12	adj→adp 12	213	2.0K
tr	noun→adj 54	propn→noun 52	noun→verb 37	noun→propn 35	adv→adj 31	491	10.0K
vi	noun→verb 201	verb→noun 152	noun→adj 151	verb→adj 140	verb→x 83	1452	12.0K
wo	noun→propn 71	verb→noun 57	pron→det 46	noun→verb 38	verb→aux 30	640	10.4K

Table 3: Top 5 most common errors and their number of occurrences for each treebank. Also shown are the total number of errors and token count for each treebank.

what the BiLSTM and embedding layers learn about UPOS tags without adding new information. Clearly more training would likely improve the parsers fine-tuned for tagging, but it would be less clear if that would be extracting information the parser previously learnt or adding more information via MLP weights.

Figure 1 shows the average cross-over of specific error occurrences for the two systems, where only 38% of the tagger’s errors don’t occur for the parser. Table 2 shows the breakdown of errors from each system by word type class for all treebanks. The ratio of the errors is substantially different for each class: 0.42 for *open*, 0.66 for *closed*, 0.78 for

other. This perhaps suggests that the parser has a tendency to learn more syntactically fixed word types than open types. Table 4 shows the F1-score for each UPOS for both systems. For the most part the parser is pretty close to the tagger for open class tags, except for INTJ which the parser never predicts, PROPEN (32.7 less for the parser), and to a lesser extent ADJ (13.0 less). Table 3 shows the top 5 most common errors per treebank for the normal taggers where PROPEN appears in 15 error types and ADJ appears in 19 out of 55. This prevalence combined with the parsers’ poor performance for these tags suggests that errors containing these tags are especially impactful for parsers when using predicted UPOS. However, it could also be that the parsers perform poorly on predicting PROPEN tags as they occur in similar syntactic roles as NOUN tokens and as such aren’t as important for syntactic analysis.

For the closed class type tags, again the parser performs similarly to the tagger but obtains a few points less except for DET, NUM, PART, and PRON with drops for parser scores of 7.9, 15.8, 13.6, and 23.9, respectively. However, of these 4 tags, only PRON and DET appear in the most common errors and only twice and once, respectively. The most common tag to appear in an error is NOUN occurring 41 times, but there is less than one point in difference between the tagger’s performance and the parser’s for NOUN. Of these 41 appearances, 14 co-occur with ADJ and 15 with PROPEN with a fairly even split of mis-tagging NOUN as either of these tags or the other way around. So generally NOUN tokens are fairly easy to tag, but the times where the tagger fails are typically where there is confusion with ADJ and PROPEN tags. Figure 2 shows statistical metrics of the taggers’ errors. First we show the proportion of out-of-vocabulary (OOV) word

	F1-score		Tokens	Class
	Tagger	Parser		
PUNCT	99.94	99.93	19.9K	Other
SYM	97.83	0.00	0.2K	
X	76.37	54.51	2.6K	
ADJ	87.98	74.98	9.4K	Open
ADV	93.94	89.97	8.5K	
INTJ	40.91	0.00	0.1K	
NOUN	95.49	94.63	43.7K	
PROPEN	90.21	57.49	9.0K	
VERB	94.80	94.05	21.5	
ADP	97.77	94.14	9.9K	Closed
AUX	96.37	93.65	6.8K	
CCONJ	96.30	94.29	7.3K	
DET	94.73	86.88	4.2K	
NUM	93.96	78.12	4.4K	
PART	90.49	76.88	1.7K	
PRON	96.31	72.46	6.0K	
SCONJ	93.15	91.22	3.2K	

Table 4: F1-score for separate tags clustered by word type class with "Other" at the top, "Open" in the middle, and "Closed" at the bottom for all tokens in the collection of treebanks used. Also reported are the total number of tokens for each tag type present across all treebanks (Tokens).

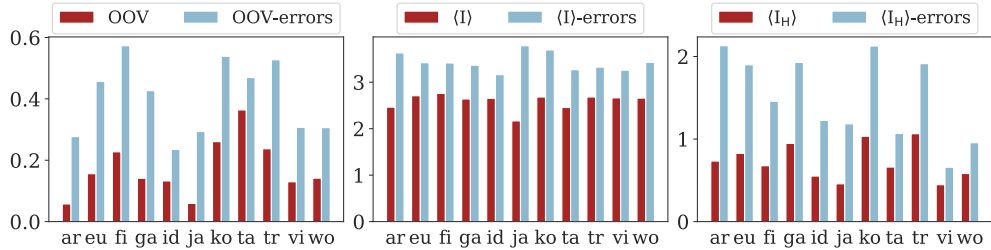


Figure 2: Measurements of all tags (red) and error (blue) tags for OOV proportion, POS bigram surprisal ($\langle I \rangle$, $\langle I \rangle$ -errors), and head POS and relation surprisal ($\langle I_H \rangle$, $\langle I_H \rangle$ -errors).

forms for all tokens and also the tokens where the tagger makes an error. Consistently across all treebanks the OOV proportion is considerably higher for tokens erroneously tagged. Second we report the mean UPOS surprisal. For a given UPOS tag, θ_n for token n , the surprisal of that UPOS tag in a given context, c_k is given as:

$$I(\theta_n) = -\log_2 p(\theta_n | c_k) \quad (1)$$

where we use a bigram context:

$$c_k = (\theta_{n-2}, \theta_{n-1}) \quad (2)$$

Then the mean surprisal, $\langle I \rangle$, over a sample of tokens is given as:

$$\langle I \rangle = \frac{1}{N} \sum_{n \in N} I(\theta_n) \quad (3)$$

where N is the number of tokens in the sample. Again, the mean tag surprisal is substantially different across all treebanks for the tokens where the tagger makes a mistake in comparison to the average over the entire treebank. Finally we report the mean surprisal of UPOS but with the context of its head’s tag and the syntactic relation joining the two tokens, such that c_k is defined as:

$$c_k = (\theta_{head}, rel) \quad (4)$$

The difference between the error sub-sample and the whole treebank is starker for the head-relation surprisal, suggesting that the tagger struggles more when the syntactic structure is uncommon.

Experiment 2: Masked tags Table 5 shows the labelled attachment scores for parsers with varying types of UPOS input. First we use the predicted output from the normal taggers from Experiment 1 (Pred) and unlike Anderson and Gómez-Rodríguez (2020) we observe a slight increase over using no

UPOS tags. However, using predicted tags isn’t universally beneficial. Arabic, Indonesian, Japanese, and Tamil all perform better with no tags.

We then used gold standard tags but masking the tags that the taggers correctly predicted to test if the erroneous tags are particularly useful. We did this for the normal taggers ($M \rightarrow E_T$) and also for the fine-tuned parsers ($M \rightarrow E_P$). The average increase for both is about 2.5 over the no tag baseline and over 2 points better than using predicted tags. Also, the improvement is universal with at least a small increase in performance over using predicted UPOS tags. Interestingly the smaller set from the tagger outperforms the larger set from the parser by 0.15, suggesting that what both the taggers and the parsers fail to capture is more important than the errors unique to the parsers. We then masked the errors from the taggers ($M \forall E_T$) to test if avoiding adding errors would still be beneficial. The performance is almost 2 points better than using the

	None	Pred.	$M \rightarrow E_T$	$M \rightarrow E_P$	$M \forall E_T$	Gold
ar	83.29	82.87	84.17	84.06	84.45	84.73
eu	81.12	81.14	82.33	82.62	83.13	84.45
fi	85.96	86.04	86.88	87.09	87.61	88.80
id	79.04	78.95	82.20	82.69	81.08	82.95
ga	76.13	76.57	76.62	76.65	77.46	77.90
ja	93.15	92.72	94.41	94.38	94.39	95.30
ko	85.40	85.86	87.53	87.82	87.44	88.52
ta	65.61	64.50	70.24	66.67	66.01	71.95
tr	66.67	67.68	67.62	67.66	67.84	68.86
vi	58.43	60.09	65.42	66.75	65.18	70.87
wo	77.87	78.49	82.03	81.39	81.11	85.41
avg	77.52	77.72	79.95	79.80	79.61	81.79

Table 5: LAS parser performance with no tags (None), with predicted tags (Pred), gold standard tags but with all tags masked except those the respective taggers predicted wrong ($M \rightarrow E_T$), similarly for the tagging errors from the fine-tuned parser ($M \rightarrow E_P$), masking the errors from the tagger ($M \forall E_T$), and finally using all gold standard tags.

predicted tags and again an increase is observed for all treebanks. This could be of use, as it is easy to envisage a tagger which learns to predict tags when a prediction is clear and to predict nothing when the probability is low. Finally, using gold standard tags is nearly 2 points better on average than the best masked tag model, which suggests that to fully utilise the information in the final few percentage that taggers miss, the full set of easy to predict tags are needed.

4 Conclusion

We have presented results which suggest that parsers do learn something of word types and that what taggers fail to learn is needed to augment that knowledge. We have evaluated the nature of typical tagging errors for a diverse subset of UD treebanks and highlighted consistent error types and also what statistical features they have compared to the average measurement across all tokens in a treebank. We have shown that it would be more beneficial to implement taggers to not only predict tags but also decide when to do so, as the errors undermine anything gained from using predicted tags for dependency parsers. Note that while we only used one parser system, the original paper (Anderson and Gómez-Rodríguez, 2020) which prompted this work observed similar behaviour with regard to predicted UPOS tags for both the system used here (graph-based) and a neural transition-based parser, suggesting that the results discussed here might extend to other parsing systems. And while it is true that we have only investigated one POS tagger system, we feel we have been careful in not making egregiously grand claims of the universality of our findings: it is merely one data point to be considered amongst many.

Acknowledgments

This work has received funding from the European Research Council (ERC), under the European Union’s Horizon 2020 research and innovation programme (FASTPARSE, grant agreement No 714150), from MINECO (ANSWER-ASAP, TIN2017-85160-C2-1-R), from Xunta de Galicia (ED431C 2020/11), and from Centro de Investigación de Galicia “CITIC”, funded by Xunta de Galicia and the European Union (ERDF - Galicia 2014-2020 Program), by grant ED431G 2019/01. The authors would also like to thank the reviewers for their suggestions and criticisms.

References

- Ramadan Alfared and Denis Béchet. 2012. Pos taggers and dependency parsing. *International Journal of Computational Linguistics and Applications*, 3(2):107–122.
- Mark Anderson and Carlos Gómez-Rodríguez. 2020. On the frailty of universal POS tags for neural UD parsers. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 69–96.
- Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. *arXiv preprint arXiv:1508.00657*.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Mary Dalrymple. 2006. How much can part-of-speech tagging help parsing? *Natural Language Engineering*, 12(4):373–389.
- Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. *Proceedings of the 5th International Conference on Learning Representations*.
- Timothy Dozat, Peng Qi, and Christopher D Manning. 2017. Stanford’s graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30.
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.
- Miryam de Lhoneux, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. 2017. From raw text to universal dependencies-look, no tags! In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 207–217.
- Aaron Smith, Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2018. An investigation of the interactions between pre-trained word embeddings, character models and pos tags in dependency parsing. *arXiv preprint arXiv:1808.09060*.
- Clara Vania, Yova Kementchedjheva, Anders Søgaard, and Adam Lopez. 2019. A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1105–1116, Hong

Kong, China. Association for Computational Linguistics.

Atro Voutilainen. 1998. Does tagging help parsing?: a case study on finite state parsing. In *Proceedings of the International Workshop on Finite State Methods in Natural Language Processing*, pages 25–36. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, et al. 2020. Universal Dependencies 2.6. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Yu Zhang, Zhenghua Li, Houquan Zhou, and Min Zhang. 2020. Is pos tagging necessary or even helpful for neural dependency parsing? *arXiv preprint arXiv:2003.03204*.

Investigation of Transfer Languages for Parsing Latin: Italic Branch vs. Hellenic Branch

Antonia Karamolegkou and Sara Stymne

Department of Linguistics and Philology
Uppsala University

Sweden

antoniakrm16@gmail.com , sara.stymne@lingfil.uu.se

Abstract

Choosing a transfer language is a crucial step in cross-lingual transfer learning. In much previous research on dependency parsing, related languages have successfully been used. However, when parsing Latin, it has been suggested that languages such as ancient Greek could be helpful. In this work we parse Latin in a low-resource scenario, with the main goal to investigate if Greek languages are more helpful for parsing Latin than related Italic languages, and show that this is indeed the case. We further investigate the influence of other factors including training set size and content as well as linguistic distances. We find that one explanatory factor seems to be the syntactic similarity between Latin and Ancient Greek. The influence of genres or shared annotation projects seems to have a smaller impact.

1 Introduction

There have been multiple projects exploiting the benefits of multilingual dependency parsing¹ (Ammar et al., 2016; Ponti et al., 2018) and especially the use of transfer learning in low-resource scenarios (Guo et al., 2015; Ponti et al., 2018). Transfer learning in the context of parsing low-resource languages uses knowledge from a transfer language in order to parse the low-resource target language (Pan and Yang, 2010). Determining the optimal transfer language for any target language is a crucial step usually leading to the selection of a language that belongs to the same language family as the target language (Dong et al., 2015; Guo et al., 2016; Dehouck and Denis, 2019). However, language proximity is not always the best criterion, since there are other properties that

could lead to better results such as the content of the syntactic, geographical, or phonological distances, which is confirmed by studies both in Machine Translation (Bjerva et al., 2019) and Syntactic Parsing (Lin et al., 2019). Smith et al. (2018) noted that for Latin, it was useful to group it with other ancient languages such as ancient Greek and Gothic, but they did not provide a comparison with other potential transfer languages.

We perform an investigation of parsing Latin in a low-resource setting, with the goal of investigating if Greek languages are better as transfer languages than Italic languages. We also explore the role of factors such as treebank size, treebank content and linguistic distance measures. We find that ancient Greek, and also modern Greek, are indeed a better choice as transfer languages for Latin than the related Italic languages Italian and French. We further show that while using ancient Greek data from the same annotation project is preferable, it is not the sole cause of the strong results, since good results are had also across different annotation projects. These results also hold for different training data sizes. Finally we note that ancient Greek is syntactically more similar to Latin than Italian, which can be an explanatory factor.

2 Related Work

Multilingual parsing has been an active topic of research over the last decade, but there is a limited number of studies that focus on transfer language selection. There are works that include language selection techniques for dependency parsing such as using a typological database to choose transfer languages based on their typological weight similarities to the target language (Søgaard and Wulff, 2012). Similarly, Agić (2017) use a part-of-speech sequence similarity method between the source and target language. A more detailed investigation on transfer language selection is performed by Lin et al. (2019). They attempt to build

¹often mentioned as cross-lingual dependency Parsing

models that rank languages based on linguistic distance measures in order to predict the optimal transfer languages. Another option is to choose the most suitable single-source parser among a set of parsers, either at the level of language (Rosa and Žabokrtský, 2015) or for individual sentences (Litschko et al., 2020), often based on part-of-speech patterns.

3 Experimental Setup

Our main aim is to investigate the impact of different transfer languages on low-resource Latin parsing. In addition we explore the impact of training data size and content, as well as the connection to a number of distance measures between languages.

3.1 Parser

To train and evaluate the parsing models we use UUParser² (de Lhoneux et al., 2017). It is a transition-based parser using a two-layer BiLSTM to extract features, and a multi-layer perceptron to predict transitions. Words are represented by a word embedding, a character embedding and a treebank embedding. Treebank embeddings represent the source treebank of each token, and has been shown to be effective both in a multilingual (Smith et al., 2018) and monolingual (Stymne et al., 2018) settings. An arc-hybrid transition system with a swap transition and a static-dynamic oracle (de Lhoneux et al., 2017) is used. It can handle non-projectivity, which is quite common in Latin.

We keep the default hyperparameter settings of the parser from Smith et al. (2018). All embeddings are initialized randomly at training time. For evaluation, we use Labeled Attachment Score (LAS). All models are trained for 30 epochs. The best epoch is selected according to the best average development set LAS score.

3.2 Language and Treebank selection

Latin is used as the target/low-resource language and we choose two transfer languages from each language family. The languages from the Italic branch, Italian and French, belong to a branch with languages historically evolved from Latin and are relatively closely related to the target language. Ancient Greek and its descendant language, modern Greek, on the other hand, belong to the Hellenic branch of the Indo-European Languages, and

²<https://github.com/UppsalaNLP/uuparser>

these languages are not as closely related as languages from the Italic branch (Nordhoff and Hammarström, 2011; Dehouck and Denis, 2019).

We use corpora from the Universal Dependencies (UD) project (Nivre et al., 2020) version 2.5 (Zeman et al., 2019). The data is sampled by choosing the first n sentences from each treebank. In two cases the Latin and ancient Greek datasets come from the same annotation projects. The Perseus treebanks have parallel texts from the Bible and classical writers (Bamman and Crane, 2011), while the PROIEL treebanks have similar texts from the new testament, but they also include texts from different authors (Haug and Jøhndal, 2008). Both the text overlap and supposedly similar annotation styles between these treebanks have been hypothesized as one possible cause of the fact that combining Latin and ancient Greek is useful (Smith et al., 2018).

We also want to investigate the effect of the size of training data, both for the target and transfer treebanks. For the target treebank, where we focus on a low-resource scenario, we use 250 and 500 sentences, respectively, while we use 2.5K and 10K sentences for the transfer languages. In the latter scenario we focus on Italian and ancient Greek, due to the small size of the modern Greek treebank and the poor performance with French as a target language. Table 1 contains information about the treebanks. All development and test sets include 250 sentences.

3.3 Linguistic Distances

Linguistic distance defines how distant a set of languages is based on genealogical, geographical, or typological features created with linguistic analysis (Lin et al., 2019). Littell et al. (2017) provide various vector information on linguistic features in URIEL Typological database which can be used to calculate how distant are the languages.³ In this work using the URIEL database we use the following linguistic distances:⁴

- **Geographic distance** (d_{geo}): The spherical distance among languages on Earth’s surface, divided by the diametrically opposite Earth’s distance. The language points are abstractions, and not precise facts, derived from

³<https://github.com/antonisa/lang2vec>

⁴Inventory distance was not used in this study, since it is similar to phonological distance, but the phonological feature vectors are derived from PHOIBLE database

Language	Treebank	Size	Genre	Exp1	Exp2	Exp3
Latin	la_Perseus	2,273	Bible, Classical texts	250	500	500
	la_proiel	18,411	New Testament, Classical texts	250	500	500
	la_ittb	26,977	Classical texts	250	500	500
Italian	it_isdt	14,167	News, legal, wiki	2,500	2,500	10,000
	it_vit	10,087	News, Politics, Literary	2,500	2,500	10,000
Ancient Greek	grc_Perseus	13,919	Bible, Classical texts	2,500	2,500	10,000
	grc_proiel	17,080	New testament, Classical texts	2,500	2,500	10,000
Modern Greek	el_gdt	2,521	News, Politics, Health	2,500	2,500	–
French	fr_ftb	18,535	News, Politics	2,500	2,500	–

Table 1: Treebank information and the number of sentences used in each experiment.

	d _{geo}	d _{gen}	d _{fea}	d _{pho}	d _{syn}
Italian	0.0	0.5	0.7	0.2	0.52
French	0.1	0.68	0.8	0.54	0.71
ancient Greek	0.0	0.8	0.3	0.2	0.35
modern Greek	0.1	1	0.8	0.59	0.64

Table 2: Distances between Latin and the other languages according to the URIEL typological database

existing databases with declarations on language location (Littell et al., 2017).

- **Genetic distance** (d_{gen}): The genealogical distance among languages, according to the hypothesized world language family tree in the Glottolog catalogue (Nordhoff and Hammarström, 2011).
- **Phonological distance** (d_{pho}): The cosine distance among the phonological vectors extracted from the World Atlas of Language Structure (WALS) and Ethnologue databases (Dryer and Haspelmath, 2013; Lewis, 2009).
- **Syntactic distance** (d_{syn}): The cosine distance among vectors mostly extracted from the syntactic structures of the languages according to WALS (Dryer and Haspelmath, 2013).
- **Featural distance** (d_{fea}): The cosine distance between feature vectors from a combination of the linguistic features described above (geographic, genetic, syntactic, phonological, inventory) extracted from the URIEL database.

All the leveraged information from the URIEL database can be found in Table 2, where the values range from 0.0 to 1.0.; numbers close to 0.0 represent proximity and vice versa. The language codes are based on the ISO-639-3 codes.⁵ In order to examine whether these linguistic distances are related to the parsing results, the Pearson Correlation Coefficient will be used.

⁵<https://iso639-3.sil.org/code/ables/639/data>

4 Results

Table 3 shows results from training a monolingual model for each Latin treebank with a small amount of data. As expected, the scores are quite low, given the limited training data size, but there is a large improvement from doubling the data from 250–500 sentences of up to 8.4 LAS points. There is a large difference in performance between the treebanks, where the Persues treebank seems to have the most challenging test set.

Table 4 shows results with a cross-lingual model with 2.5K transfer language sentences and Table 5 shows the results with 10K transfer language sentences. In all cases, one of the ancient Greek treebanks give the best results, with improvements of up to 16.9 LAS points compared to the monolingual baseline for Latin PROIEL. In all but one case, modern Greek also surpasses the results of all Italic treebanks, and also beats all monolingual baselines. Italian helps for the PROIEL and ITTB Latin treebanks, but in most cases hurts slightly for the Persues treebank. French, on the other hand leads to very poor results in all cases, mostly giving worse results than the monolingual baseline.

Concerning the impact of training data size, we can usually see a large improvement, when doubling the target data, just as in the monolingual case. Overall the improvements are larger for the poor models than for the stronger ones. Increasing the size of the transfer language from 2.5K to 10K further improves the results in most cases when ancient Greek is used as transfer language. The improvements are typically smaller than when increasing the size of the target language, though. When using Italian as the transfer language, however, the results do not show much change compared to using less Italian data, sometimes even leading to worse results. It thus seems that using more data from the transfer language is only useful for transfer languages that are a good fit to the

Training sentences:	250	500
la_Perseus	17.9	26.1
la_proiel	39.9	43.1
la_ittb	33.1	41.6

Table 3: LAS scores for monolingual training with 250 and 500 sentences.

Target sent.	la_Perseus		la_proiel		la_ittb	
	250	500	250	500	250	500
it_isdt	19.9	25.9	46.5	55.6	38.1	46.4
it_vit	17.8	24.5	44.2	54.7	36.9	44.3
grc_Perseus	30.1	32.4	50.4	58.1	39.9	45.4
grc_proiel	27.6	31.9	50.9	60	40.4	47.6
el_gdt	23.6	27.2	48.5	58.4	36.6	46.6
fr_ftb	12.8	22.8	39.8	50.3	13.7	40.2

Table 4: LAS scores for multilingual experiments with 2.5K sentences from the transfer language, and 250 or 500 sentences from the target language.

target language.

For Latin PROIEL and Perseus, where there are ancient Greek treebanks from the corresponding annotation projects, it is always preferable to use the matching treebank. However, the gaps are typically not large, ranging from 0.4 to 3.9 LAS points, with the scores for the non-matching treebank in most cases beating the scores for treebanks from all other languages. Also for the Latin ITTB treebank, the scores for both non-matching ancient Greek treebanks are among the highest scores, with the PROIEL treebank being the best match. This indicates that the impact of annotation project, with content and annotation styles matching, adds to the performance, but is not the main explanatory factor for the usefulness of ancient Greek. It is also worth noting that the treebanks for Italian VIT, modern Greek and French have similar content, but very different parsing results, indicating that language choice is more important than the genres of the treebanks.

Table 6 shows Pearson correlations between the distance measures and the parsing scores for the Latin PROIEL treebank using 500 sentences and 2.5K transfer language sentences. There is a strong negative correlation of -0.76 between the syntactic distance of the languages and the parsing results, even though it is not significant. This finding seems reasonable since syntactic features of a language are intuitively important for parsing. Ancient Greek and Latin actually have a closer syntactic distance than Italian and Latin, see Table 2. The same applies to the featural distance, which is

	la_Perseus	la_proiel	la_ittb
it_isdt	24.3	55.3	44.7
it_vit	24	55.4	42.7
grc_Perseus	36.9	60.7	46.6
grc_proiel	33	62.3	47.3

Table 5: LAS scores from multilingual experiments with 10K sentences from the transfer language and 500 from the target language

	R	Strength	P-value
d _{geo}	-0.47	weak	0.34
d _{gen}	0.57	moderate	0.23
d _{fea}	-0.91	strong	0.011
d _{pho}	-0.44	weak	0.382
d _{syn}	-0.76	strong	0.073

Table 6: Pearson correlation and p-value between parsing scores and linguistic distance measures for the Latin PROIEL treebank.

a combination of various features (including syntactic, phonological, inventory, geographic, and genealogical), and has a strong significant negative correlation of -0.91 . While this finding is quite intuitive, it is contrary to the finding of Lin et al. (2019) who found that geographic and genetic distances were more important than syntactic or featural distance, however, for 0-shot parsing with a higher number of languages. It is, however, in accordance with (Bjerva et al., 2019) who indicated that structural similarity is a better predictor of language representation similarities compared to genetic similarity. The strong performance for Hellenic languages is especially interesting since they do not share script with Latin, which means that the character embeddings in UUParser are less useful than for Italian.

5 Conclusion

We have shown that using Hellenic languages is preferable to using Italic languages when training a multilingual parsing model for Latin in a low resource scenario. While we see the best results when we use ancient Greek treebanks from the same annotation project as the Latin treebanks, we also see very competitive results when training across annotation projects, mostly surpassing all other languages explored. We also see that it is more useful to increase the training data size of the target language than the transfer language, and that increasing the size of the target language is only useful when it is a good match. Finally we show that there are strong correlations between the parsing result and the featural and syntactic dis-

tance of the target and transfer language, which could explain the usefulness of ancient Greek, the most syntactically similar language to Latin in our sample.

In this study we only explored a low-resource setting, using a limited amount of Latin data. It would be interesting to see if the findings hold also when we use all available data, as indicated by the results of Smith et al. (2018). We would also like to add pre-trained word embeddings, either cross-lingual static embeddings, or multilingual contextual embeddings, to see what the impact is, compared to our current experiments where we do not use any pre-trained embeddings. Another direction would be to investigate if ancient Greek is a good transfer language for Latin also for other tasks, which might be less sensitive to syntactic distance.

References

- Željko Agić. 2017. Cross-lingual parser selection for low-resource languages. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 1–10, Gothenburg, Sweden. Association for Computational Linguistics.
- Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.
- David Bamman and G. Crane. 2011. The Ancient Greek and Latin dependency treebanks. In *Language Technology for Cultural Heritage*.
- Johannes Bjerva, Robert Östling, Maria Han Veiga, Jörg Tiedemann, and Isabelle Augenstein. 2019. What do language representations really represent? *Computational Linguistics*, 45(2):381–389.
- Mathieu Dehouck and Pascal Denis. 2019. Phylogenetic multi-lingual dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 192–203, Minneapolis, Minnesota. Association for Computational Linguistics.
- Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.
- Matthew S Dryer and Martin Haspelmath. 2013. *The World Atlas of Language Structures Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology.
- Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1234–1244, Beijing, China. Association for Computational Linguistics.
- Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2016. A representation learning framework for multi-source transfer parsing. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI’16*, page 2734–2740. AAAI Press.
- Dag T. T. Haug and Marius L. Jøhndal. 2008. Creating a parallel treebank of the old indo-european bible translations. In *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH)*, pages 27–34.
- M. Paul Lewis, editor. 2009. *Ethnologue: Languages of the World*, sixteenth edition. SIL International, Dallas, Texas, USA.
- Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2017. Arc-hybrid non-projective dependency parsing with a static-dynamic oracle. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 99–104, Pisa, Italy. Association for Computational Linguistics.
- Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.
- Robert Litschko, Ivan Vulić, Željko Agić, and Goran Glavaš. 2020. Towards instance-level parser selection for cross-lingual transfer of dependency parsers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3886–3898, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.
- Sebastian Nordhoff and Harald Hammarström. 2011. Glottolog/langdoc: Defining dialects, languages, and language families as collections of resources. In *Proceedings of the First International Workshop on Linked Science 2011*, volume 783 of *CEUR Workshop Proceedings*.
- S. J. Pan and Q. Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.
- Edoardo Maria Ponti, Roi Reichart, Anna Korhonen, and Ivan Vulić. 2018. Isomorphic transfer of syntactic structures in cross-lingual NLP. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1531–1542, Melbourne, Australia. Association for Computational Linguistics.
- Rudolf Rosa and Zdeněk Žabokrtský. 2015. KLcpos3 - a language similarity measure for delexicalized parser transfer. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 243–249, Beijing, China. Association for Computational Linguistics.
- Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018. 82 treebanks, 34 models: Universal dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.
- Anders Søgaard and Julie Wulff. 2012. An empirical study of non-lexical extensions to delexicalized transfer. In *Proceedings of COLING 2012: Posters*, pages 1181–1190, Mumbai, India. The COLING 2012 Organizing Committee.
- Sara Stymne, Miryam de Lhoneux, Aaron Smith, and Joakim Nivre. 2018. Parser training with heterogeneous treebanks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 619–625, Melbourne, Australia. Association for Computational Linguistics.
- Daniel Zeman, Joakim Nivre, Mitchell Abrams, et al. 2019. Universal dependencies 2.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Toward cross-lingual application of language-specific PoS tagging schemes

Hinrik Hafsteinsson and Anton Karl Ingason

University of Iceland

Reykjavík, Iceland

{hinhaf, antoni}@hi.is

Abstract

We describe the process of conversion between the PoS tagging schemes of two languages, the Icelandic MIM-GOLD tagging scheme and the Faroese Sosialurin tagging scheme. These tagging schemes are functionally similar but use separate ways to encode fine-grained morphological information on tokenised text. As Faroese and Icelandic are lexically and grammatically similar, having a systematic method to convert between these two tagging schemes would be beneficial in the field of language technology, specifically in research on transfer learning between the two languages. As a product of our work, we present a provisional version of Icelandic corpora, prepared in the Faroese PoS tagging scheme, ready for use in cross-lingual NLP applications.

1 Introduction

Part of Speech (PoS) tagging is the process of labelling words and symbols of running text based on their lexical category and morphological features. Text corpora that have been PoS-tagged in this way serve as a valuable tool in various fields of linguistic research and language technology. The specifics and format of the PoS tags used, the tagging scheme, varies greatly between languages and applications. In the current project, we focus on two languages with significant linguistic similarities, Icelandic and Faroese, and PoS tagging schemes for the two which overlap significantly in function; the MIM-GOLD tagging scheme (Barkarson et al., 2020) and the Sosialurin tagging scheme (Hansen et al., 2004), respectively.

Icelandic and Faroese are distinct yet relatively similar languages, with their similarities especially apparent in morphology and syntax. While

Icelandic has seen significant gains in the field of language technology (LT) over the past few decades (Nikulásdóttir et al., 2017), the same is not true for Faroese. Due to the similarities between the two, there is a real possibility that employing transfer learning, using Icelandic data in tandem with Faroese, to create effective LT tools and digital language resources for Faroese.

With the end goal of cross-lingual transfer learning in mind, we focus on the task of PoS tagging. Our goal is to produce an effective way to map between the tagging schemes used for the two languages. This requires some revisions to one of the tagging schemes and assurance that a one-to-one mapping between tagsets is possible.

The paper is structured as follows. Section 2 discusses the possibilities of cross-lingual transfer learning between Faroese and Icelandic. Section 3 describes the Icelandic MIM-GOLD tagging scheme and Section 4 the Faroese Sosialurin tagging scheme. Section 5 discusses the current differences between the two tagging schemes and Section 6 details the procedure of converting between the two tagsets, while Section 7 discusses possible alternatives such as conversion. Section 8 concludes.

2 Faroese, Icelandic and transfer learning

The fundamental reason that makes Icelandic NLP implementations applicable for Faroese are the grammatical similarities between the two languages. These similarities are especially apparent in morphology, as both languages retain grammatical categories not apparent in other similar languages, e.g., four grammatical cases for nominals and an extensive conjugation system for verbs, to name a few. Furthermore, the similarities also extend to the syntax of the languages and orthographies, although with various systematic differences in both. With this in mind it can be sup-

posed that NLP solutions that perform well for Icelandic may also perform well for Faroese, especially data-driven applications.

Some data already exists on the efficacy of cross-lingual transfer learning between Icelandic and Faroese. The FarParsald project (Ingason et al., 2014) focused on using a syntactically annotated corpus of Faroese, the Faroese Parsed Historical Corpus (FarPaHC; Sigurðsson et al. 2012), to train a syntactical parser, FarParsald, based on the data-driven Berkeley parser (Petrov et al., 2006; Petrov and Klein, 2007). The relatively small FarPaHC corpus, containing about 40,000 tokens, was supplemented with excerpts from its Icelandic counterpart, the one million word Icelandic Parsed Historical Corpus (IcePaHC; Rögnvaldsson et al. 2012). Using this approach, the overall parsing accuracy of FarParsald was raised from 75.44% to 78.06%, when 20% of the IcePaHC corpus, about 200,000 tokens, was added to the Faroese training data. In effect, a training set made of mostly Icelandic data returned better results than the Faroese-only data.

A similar approach may be taken in PoS tagging Faroese. ABLTagger (Steingrímsson et al., 2019), a recent Bi-LSTM driven PoS Tagger has shown impressive results in data-driven tagging of Icelandic. This implementation might well serve as a platform for further transfer learning between the two languages.

3 The MIM-GOLD tagging scheme

The Icelandic tagging scheme we use in our project is the MIM-GOLD tagging scheme, used in its eponymous corpus (Barkarson et al., 2020), a one million word, hand corrected corpus which serves as a gold standard for PoS tagging Icelandic. This tagging scheme is a modified version of the one used in the Icelandic Frequency Dictionary (IFD) corpus (Pind et al., 1991), with various revisions made to the tagset to improve and streamline machine tagging of texts.

In this tagging scheme, each token receives one PoS tag, consisting of a tag string. Each tag string consists of a series of characters, each having a particular morphosyntactic function, e.g., case, number, tense and grammatical gender. This is illustrated in Table 1, where the sentence in (1) is shown when tagged using the MIM-GOLD tagging scheme.

- (1) Ég stökk á eftir strætó og veifaði.
I jumped on after bus and waved
'I jumped after the bus and waved.'

Token	PoS tag	Explanation
Ég	fp1en	f : pronoun; p : personal; 1 : 1st person; e : singular; n : nominative;
stökk	sfg1eþ	s : verb; f : indicative; g : active; 1 : 1st person; e : singular; þ : past tense
á	aa	a : adverb; a : doesn't govern case;
eftir	af	a : adverb; þ : governs case;
strætó	nkeþ	n : noun; k : masculine; e : singular; þ : dative;
og	c	c : conjunction;
veifaði	sfg1eþ	s : verb; f : indicative; g : active; 1 : 1st person; e : singular; þ : past tense
.	pl	p -punctuation, l -end of sentence

Table 1: A sentence tagged with the MIM-GOLD tagging scheme, with explanations.

4 The Sosialurin tagging scheme

The Faroese PoS tagging scheme we focus on is the one used in the Sosialurin corpus, devised by Hansen et al. (2004) as part of a larger project to create a PoS-tagged corpus for the language and train automatic PoS tagging software. This scheme is, to a large extent, based on the tagging scheme used in the IFD corpus for Icelandic (Pind et al., 1991). This was possible because of the many similarities between Icelandic and Faroese in morphology and grammar in general.

As in its Icelandic counterpart, the Faroese tagging scheme assigns each token a tag string, which contains a series of letters, each signifying relevant morphosyntactic information. The languages are not identical, however, and this is reflected in the Faroese tagging scheme. Furthermore, in a handful of grammatical categories, the Sosialurin tagging scheme encodes fewer details than the Icelandic one. In short, it is not as fine grained. Finally, the tagging schemes use different symbols in the tag strings themselves, rendering the tagging schemes superficially different. An example of the Sosialurin tagging scheme in practice is shown in Table 2, where the tokens of the sentence in (2) are shown with respective PoS tags.

- (2) Hann er grivin undir Hamrum.
he is buried under Hamrar
'He is buried at Hamrar.'

As discussed in Section 3 a number of revisions have been made to the IFD tagging scheme,

Token	Tag	Explanation
Hann	PPMSN	P -pronoun, P -personal M -masculine, S -singular, N -nominative
er	VNPS3	V -verb, N -indicative, P -present, S -singular, 3 -third person
grivin	VAMSN	V -verb, A -past participle, M -masculine, S -singular, N -nominative,
undir	ED	E -preposition, D -governs dative
Homrum	SMSDL	S -noun, M -masculine, S -singular, D -dative, L -location

Table 2: Example of the Sosialurin tagset, with explanations

Pronouns:	Added subcategories to tagstring
Adverbs:	Interjections and prepositions tagged as adverbs
Numerals:	New and reorganised subcategories
Abbreviations:	Subcategories for different types of abbreviations
Verbs:	Dedicated tag for supine removed
Nouns:	Place names and names of persons merged
Other:	New dedicated classes for punctuation and e-mail/web addresses

Table 3: Revisions applied to the Sosialurin tagging scheme based on the Icelandic *MIM-GOLD*.

mostly to improve tagging efficiency, culminating in the current MIM-GOLD tagging scheme for Icelandic. The same cannot be said about the Sosialurin tagging scheme, as no substantial revisions have been made to it since its inception. As such, we suggest a set of revisions to the Sosialurin tagging scheme, largely in step with the revisions made for the MIM-GOLD tagging scheme. These revisions are listed in Table 3.

The revisions applied to the Sosialurin tagging scheme include reworked numeral and punctuation tag strings, simplified case governance tagging for adverbs and the removal of a dedicated tag for past participles. Furthermore, various new tag strings were introduced, based on features from the original IFD tagging scheme which were omitted from the original Faroese scheme, e.g., distinction between different categories of pronouns.

In addition to the MIM-GOLD based revisions, we suggest a possible language-specific revision to the Faroese taggings scheme. This entails the removal of distinction between person (1st, 2nd or 3rd) from verb tags in the original tagset. In Faroese, person is never morphologically distinct in verbal plural forms, and may thus be redundant in the tagging scheme, in theory. Such a revision would improve the accuracy of machine-tagging,

but downstream effects, e.g., on syntactic parsing, are not clear. As such, we leave it as an open suggestion and do not apply it in our project.

With all revisions applied, the total number of theoretical tags in the Sosialurin tagset is about 600. When applied to the original Sosialurin corpus, 379 of these tags appear in the corpus, while the original corpus contained 390 unique tags. This is to be expected, mostly due to the simplified punctuation tags in the revised tagging scheme.

The revisions applied to the Faroese tagging scheme have been shown to positively affect overall PoS tagging accuracy. When applied to the Sosialurin corpus and evaluated using ten-fold cross validation, a Faroese implementation of ABLTagger achieved an overall error reduction rate of 7.51% (Hafsteinsson and Ingason, 2021).

5 Remaining tagging scheme differences

With the revisions based on MIM-GOLD, described in Section 4, to the Faroese tagging scheme, the function of the two tagging schemes has become markedly more similar. The remaining aspect separating the two are language-specific features of the two schemes, specifically concerning verbal PoS tags and the interpretation of article tags.

Both Icelandic and Faroese make a morphological distinction between two voices for verbs, the active and middle voices. The MIM-GOLD tagging scheme for Icelandic treats the verbal voice as a defining characteristic of all verbs. In the tag string, this is shown with the letter *g* for the active voice, and *m* for the middle voice. However, in the Sosialurin tagging scheme for Faroese, the verbal voice is instead treated as a verbal *mood*. This causes a discrepancy between the two tagging schemes, as the hierarchy of the verbal tag string is fundamentally different. A verb in Icelandic, tagged as being in the indicative mood, could either be in the active or middle voice. This is not possible in the Faroese tagging scheme, since the middle voice is considered a verbal mood; the hierarchical nature of the tag string doesn't allow two different mood labels.

The reason for this difference might be differences in the languages themselves. Although both Faroese and Icelandic exhibit what may be called a grammatical voice in verbs, the Faroese form is likely reduced compared to the Icelandic. In turn, the distinction between voice in Faroese verbs is

not as fundamental as in Icelandic. With this in mind, the discrepancy as a whole may be tentatively circumvented in the tag conversion.

A more significant difference between the two tagging schemes concerns the article word class. The Icelandic tagging scheme tags uses a specific tag for definite articles, which reflects conventional analyses of Icelandic grammar, in which the free-standing definite article ‘hinn’ is classified as a distinct word class, with no indefinite article being used. This free-standing article is thought of as a literary device of irregular usage, with the more common suffixed definite article being in more general use. Conversely, Faroese uses both definite and indefinite free-standing articles; ‘tann’ and ‘hin’ as definite and ‘ein’ as indefinite, along with a suffixed definite article, like Icelandic (Þráinsson et al., 2004). Despite the apparent function of these words as articles within Faroese, these words are tagged as indicative pronouns in the Faroese tagging scheme, forgoing a distinct article tag altogether. Furthermore, this seems to be an inherent difference between the conventional analyses between the two languages, which discourages the approach of simply adding an article tag to the tagging scheme.

6 Conversion between tagsets

We suggest a partial solution to the effect of the inherent differences between the two tagging schemes, when converting between the two. Concerning the verbal tags, when converting from the Faroese tagging scheme to the Icelandic, all verb PoS tags *not* tagged as in the middle voice are mapped to equivalent verbal tags in the indicative mood, active voice. Faroese verbs tagged in the middle are, conversely, mapped to the indicative middle voice. The opposite is done when converting from Icelandic to the Faroese tagging scheme, with the information on mood being overwritten, in the case of verbs that are in the middle voice. This approach produces a one-to-one mapping between the two tagging schemes and mitigates the discrepancy between them. This is especially efficient when only converting from the Icelandic to Faroese, which suffices use in cross-lingual transfer learning, as described in Section 2.

Regarding the difference concerning the article class, further research is needed before an end result is settled on. The conversion between tagsets itself is not hampered by the absence of a distinct

article tag in the Faroese tagging scheme, but it may have an effect when applying datasets with converted tagging schemes, e.g., in transfer learning. Future work will shed more light on this.

With this in mind, we have set up simple Python scripts which generate full tagsets for the tagging schemes and convert between the two. Furthermore, we have produced preliminary datasets for use in testing of cross-lingual transfer learning, based on the MIM-GOLD corpus for Icelandic, the tagset of which was used in the development of the conversion described above. The conversion scripts and training datasets are tentatively made available on GitHub¹ as products of this project.

7 Alternatives to conversion

Although we the main objective of the current project concerns the conversion between two tagging schemes, we are remain aware of the possibility of alternatives to this approach. One notable possibility would be to simply unify the two tagging schemes. With the modifications described in Section 4 applied to the Faroese tagging scheme, the two tagging schemes become near identical in function. If the end goal is to align one tagging scheme to the other, it begs the question whether a single tagging scheme would suit the needs of the two languages for use in NLP, e.g. by simply using the established Icelandic MIM-GOLD tagging scheme to describe both. The grammatical similarities between the two languages, discussed in Section 2 further supports this argument. However, as the remaining discrepancies between the tagging schemes suggests, this approach is at best inopportune. At the moment, the conventional analyses of the two languages differ in such a way that simply applying the Icelandic MIM-GOLD tagging scheme on Faroese text would be sub-optimal. However, experimenting on this could be fruitful, and reconciling these differences in analysis at a future date may also be possible.

Circumventing the topic of the two tagging schemes discussed here, it should be noted that both Faroese and Icelandic have been described using the Universal Dependencies (UD) annotation scheme. Three UD corpora are available for Icelandic and two for Faroese, with considerable overlap in the production of the Faroese FarPaHC corpus and the Icelandic IcePaHC and Modern

¹https://github.com/hinrikur/far-ice_corpora

corpora, each being converted to the UD format from existing datasets, as described by Arnardóttir et al. (2020). In this sense the two languages have already been described with a common annotation framework, although the UD annotating scheme is not strictly a dedicated PoS tagging scheme compared to the two tagging schemes used in our project.

8 Conclusion

We have described the process of conversion between the PoS tagging schemes of two grammatically similar languages, the Icelandic MIM-GOLD tagging scheme and the Faroese Sosialurin tagging scheme. Despite the two tagging schemes being functionally similar, they use separate ways to encode fine-grained morphological information on tokenised text. We described the differences between the two, along with revisions made to the Faroese tagging scheme, with the goal of streamlining automatic PoS tagging. We discussed grammatical differences between Faroese and Icelandic which result in minor discrepancies between the two tagging schemes and suggested a way to mitigate the effects of this when converting between the two. As a result, we produced a simple way to convert PoS tags between the languages. The results of our work have been made available for use, consisting of Python scripts for converting Icelandic and Faroese tagged corpora and preliminary converted training data, ready for application in cross-lingual NLP applications, with the end goal of it being of benefit in cross-lingual transfer learning.

References

- Þórunn Arnardóttir, Hinrik Hafsteinsson, Einar Freyr Sigurðsson, Kristín Bjarnadóttir, Anton Karl Ingason, Hildur Jónsdóttir, and Steinþór Steingrímsson. 2020. <https://www.aclweb.org/anthology/2020.udw-1.3> A Universal Dependencies conversion pipeline for a Penn-format constituency treebank. In *Proceedings of the Fourth Workshop on Universal Dependencies (UDW 2020)*, pages 16–25.
- Starkaður Barkarson, Einar Freyr Sigurðsson, Eiríkur Rögnvaldsson, Hildur Hafsteinsdóttir, Hrafn Loftsson, Steinþór Steingrímsson, and Þórdís Dröfn Andrésdóttir. 2020. <http://hdl.handle.net/20.500.12537/39> MIM-GOLD 20.05. CLARIN-IS, Stofnun Árna Magnússonar.
- Hinrik Hafsteinsson and Anton Karl Ingason. 2021. Shared digital resource application within insular scandinavian. In *Proceedings of the Workshop on Computational Methods for Endangered Languages*, volume 1, pages 70–79.
- Zakaris Svabo Hansen, Heini Justinussen, and Mortan Ólason. 2004. Marking av teldutökum tekstsavni [Tagging of a digital text corpus].
- Anton Karl Ingason, Hrafn Loftsson, Eiríkur Rögnvaldsson, Einar Freyr Sigurðsson, and Joel Wallenberg. 2014. Rapid Deployment of Phrase Structure Parsing for Related languages: A Case Study of Insular Scandinavian. In *Proceedings of Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, pages 91–95.
- Anna Björk Nikulásdóttir, Jón Guðnason, and Steinþór Steingrímsson. 2017. *Mál tækni fyrir íslensku 2018–2022: verkáætlun [Language Technology for Icelandic 2018-2022: Strategic Plan]*.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440.
- Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411.
- Jörgen Pind, Friðrik Magnússon, and Stefán Briem. 1991. *Íslensk orðtíðnibók [The Icelandic Frequency Dictionary]*. The Institute of Lexicography, University of Iceland, Reykjavík, Iceland.
- Eiríkur Rögnvaldsson, Anton Karl Ingason, Einar Freyr Sigurðsson, and Joel Wallenberg. 2012. The Icelandic Parsed Historical Corpus (IcePaHC). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 1977–1984.
- Einar Freyr Sigurðsson, Anton Karl Ingason, Eiríkur Rögnvaldsson, and Joel C. Wallenberg. 2012. <http://www.linguist.is/farpahc> Faroese Parsed Historical Corpus (Far PaHC). Version 0.1.
- Steinþór Steingrímsson, Örvar Káráson, and Hrafn Loftsson. 2019. Augmenting a BiLSTM Tagger with a Morphological Lexicon and a Lexical Category Identification Step. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1161–1168.
- Höskuldur Þráinsson, Hjalmar P. Petersen, Jógvan í Lón Jacobsen, and Zakaris Svabo Hansen. 2004. *Faroese: An overview and reference grammar*. Føroya fróðskaparfelag, Torshavn.

Exploring the Importance of Source Text in Automatic Post-Editing for Context-Aware Machine Translation

Chaojun Wang¹ Christian Hardmeier^{2,3} Rico Sennrich^{4,1}

¹School of Informatics, University of Edinburgh

²Department of Computer Science, IT University of Copenhagen

³Department of Linguistics and Philology, Uppsala University

⁴Department of Computational Linguistics, University of Zurich

zippo_wang@foxmail.com, chrha@itu.dk, sennrich@cl.uzh.ch

Abstract

Accurate translation requires document-level information, which is ignored by sentence-level machine translation. Recent work has demonstrated that document-level consistency can be improved with automatic post-editing (APE) using only target-language (TL) information. We study an extended APE model that additionally integrates source context. A human evaluation of fluency and adequacy in English–Russian translation reveals that the model with access to source context significantly outperforms monolingual APE in terms of adequacy, an effect largely ignored by automatic evaluation metrics. Our results show that TL-only modelling increases fluency without improving adequacy, demonstrating the need for conditioning on source text for automatic post-editing. They also highlight blind spots in automatic methods for targeted evaluation and demonstrate the need for human assessment to evaluate document-level translation quality reliably.

1 Introduction

Neural machine translation (NMT) has significantly improved the state of the art in MT (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017) on the sentence level. However, accurate translation requires looking at larger units than individual sentences (Hardmeier, 2014), and context-aware NMT has recently become a popular research direction (Miculicich et al., 2018; Scherrer et al., 2019; Junczys-Dowmunt, 2019).

One approach to discourse-level processing in NMT is automatic post-editing of the output of a sentence-level system. DocRepair (Voita et al., 2019a) is a monolingual sequence-to-sequence model to correct inconsistencies in groups of adja-

cent sentence-level translations, showing improvements for specific discourse-level phenomena such as the generation of inflections in elliptic sentences.

The hypotheses explored in this work are (1) that the coherence of the translation can be further improved by exploiting context in the source language, and (2) that the omission of source context disproportionately affects adequacy in a way that is not measured adequately by the existing automatic evaluation procedures.

Our post-editing model is a document-level adaptation of Transference (Pal et al., 2019), a successful three-way transformer architecture from the WMT 2019 Automatic Post-Editing (APE) task (Chatterjee et al., 2019). To keep the model from over-correcting the hypothesis, we use data weighting (Junczys-Dowmunt, 2018) and a conservativeness penalty (Junczys-Dowmunt and Grundkiewicz, 2016). We evaluate on the same training and evaluation sets as Voita et al. (2019a), including a general test set validated by BLEU score and contrastive sets for several discourse phenomena.

Our experimental results confirm both hypotheses. Despite similar BLEU, human evaluation demonstrates that our Transference model significantly outperforms DocRepair in terms of adequacy, whilst both models show a comparable improvement in fluency over a baseline without APE. The automatic evaluation on discourse-specific test sets suggests that source-side information is particularly useful for predicting omitted verb phrases; however, even the targeted discourse-specific evaluation does not reflect the adequacy gain found by human evaluators. This is especially true since some of the discourse-specific test sets of Voita et al. (2019a) have a very narrow focus on problems for which source context is unlikely to help.

2 Transference

Transference (Pal et al., 2019) (Figure 1) is a multi-source transformer (Vaswani et al., 2017) architec-

ture which exploits both source src and the MT output mt to predict the reference ref . It is composed of (1) a source encoder (enc_{src}) to generate the src representation, (2) a second encoder ($enc_{src \rightarrow mt}$) which is a standard transformer decoder architecture without mask to produce the representation of mt incorporating src information, and (3) a decoder (dec_{ref}) which captures the final representation from $enc_{src \rightarrow mt}$ via cross-attention.

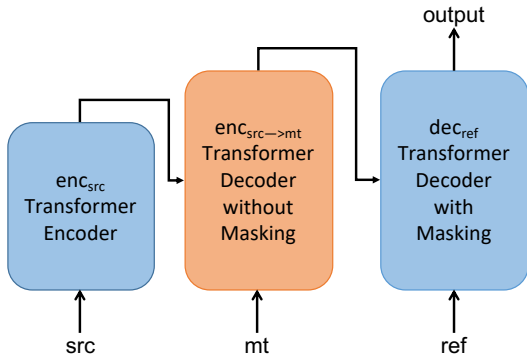


Figure 1: Transference architecture for multi-source document-level repair model.

If document-level APE is trained on a small subset of the parallel data, or only synthetic data, and therefore presumably weaker as a general model of translation than the sentence-level main model, we need to control how aggressively APE can modify mt to prevent over-correction. We adopt two strategies from the APE literature to achieve this. A *conservativeness penalty* (Junczys-Dowmunt and Grundkiewicz, 2016), denoted c , penalises the score of each prediction that is not in src or mt . Formally, let $V_c = V_{src} \cup V_{mt}$ be the subset of the full vocabulary V that occurs in an input segment. Given a $|V|$ -sized vector of candidates h_t at time step t , the score of each candidate v is defined as:

$$h_t(v) = \begin{cases} h_t(v) - c & \text{if } v \in V \setminus V_c \\ h_t(v) & \text{otherwise.} \end{cases} \quad (1)$$

Second, similar to Lopes et al. (2019), we apply a *data weighting strategy* during training. We assign each training sample a weight that is defined as $\text{BLEU}_{\text{smooth}}(mt, ref)$ (Lin and Och, 2004) to upweight samples that require little post-editing.

3 Data and Preprocessing

We use all of the English-to-Russian data released by Voita et al. (2019a)¹, including: (1) 6M context-

¹<https://github.com/lena-voita/good-translation-wrong-in-context>

Model	Deixis	Lex.c.	Ell.infl.	Ell.VP	BLEU
<i>Results reported by Voita et al. (2019a):</i>					
Baseline	50.0	45.9	53.0	28.4	32.41
DocRepair	91.8	80.6	86.4	75.2	34.60
<i>Our experiments:</i>					
DocRepair	88.6	70.5	83.8	69.0	32.69
DocRepair (+P)	87.6	67.6	82.2	71.8	32.38
Transference	86.8	62.9	81.6	73.0	30.56
Transference (+P)	87.8	65.4	84.8	82.8	32.53

Experiments marked +P use the ParData corpus.

Table 1: BLEU score on general test set and accuracy on contrastive test sets (deixis, lexical consistency, ellipsis (inflection), and VP ellipsis).

agnostic and 1.5M context-aware (4 consecutive sentences in each sample) data from the OpenSubtitles2018 corpus (Lison et al., 2018); (2) Russian monolingual data in 30M groups of 4 consecutive sentences gathered by Voita et al. (2019a). We reuse the synthetic training data for APE generated by Voita et al. (2019a), treating Russian monolingual data as ref , a sentence-level English back-translation as src , and the Russian roundtrip translation as mt . The evaluation data consists of general test sets extracted from the training data and four contrastive test sets to evaluate specific contextual phenomena.

The four contrastive test sets have a narrow focus on specific discourse-level phenomena. The “Deixis” set targets consistent use of formal and informal second-person pronouns (T-V distinction) in Russian (however without regard to the social acceptability of the selected form). “Lexical cohesion” targets the consistent transliteration of proper names into Cyrillic script. These two sets are independent of source context by design, as the model is only evaluated on the generation of consistent repetitions of a form it has committed to, regardless of its adequacy in the context. The “Ellipsis VP” set targets elliptic verb phrases, where Russian requires the production of a lexical verb form not found in English. The “Ellipsis inflection” set tests the generation of noun inflections in sentences where the governing verb has been elided.

The training data is tokenised and truecased with Moses (Koehn et al., 2007), and encoded using byte-pair encoding (Sennrich et al., 2016b) with source and target vocabularies of 32000 tokens. Like Voita et al. (2019a), we report lowercased, tokenised BLEU (Papineni et al., 2002) with *multi-bleu.perl* from the Moses toolkit.

4 Model

The sentence-level baselines (EN→RU) and model used for RU→EN back-translation are Transformer base models (Vaswani et al., 2017).

For document-level APE, DocRepair is a Transformer base model that operates on groups of adjacent sentences, mapping from *mt* to *ref*. We use the Nematus toolkit (Sennrich et al., 2017) for DocRepair and our implementation of the Transference architecture, using the same configuration as Pal et al. (2019).² Detailed hyperparameters are listed in Appendix A. We train our document-level models on the 30M pairs of synthetic data. For some models, we also include the subset of the parallel data (1.5M pairs) for which context sentences are available, referred to as *ParData*. The *mt* part of *ParData* is generated by randomly sampling 20 translations with our EN→RU baseline system.

In preliminary experiments, adding noise to the training data improved model generalisation. We generated noise with two strategies. Following Voita et al. (2019a), *mt* in both synthetic data and *ParData* is randomly selected from 20 translations, and noise is added by making random token substitutions with probability of 10%. Following Edunov et al. (2018), noise is added to the *src* in synthetic data by three operations: (1) replacing a token; (2) deleting a token; (3) swapping adjacent token pairs, with a probability of 10%.

5 Automatic evaluation

Table 1 shows the results in terms of accuracy on the contrastive test sets and BLEU on the general test set. For DocRepair, we were unable to replicate the exact results of Voita et al. (2019a). Our conclusions are based on our own implementation.

On the general test set, trained on only synthetic training data, Transference achieves about 2 BLEU points less than DocRepair. We suspect that this derives from the mismatch of the training and test data for Transference. Specifically, during training, the “source” seen by Transference is the result of noisy back-translation from Russian, whereas at test time, the source is an original English sentence. When *ParData* is included, Transference and DocRepair achieve comparable BLEU.

In accuracy on the test sets for T/V pronouns (“deixis”) and transliteration consistency (“lexical

²Code available at <https://github.com/zippotju/Context-Aware-Bilingual-Repair-for-Neural-Machine-Translation>

cohesion”), Transference does not improve over DocRepair, which is unsurprising considering how those test sets are constructed. However, adding source knowledge does improve results on both ellipsis test sets, for VP ellipsis even without adding the *ParData* data. The improvement is generally greater for VP ellipsis than for noun inflection.

6 Human evaluation

To gain a better picture of the merits of the different systems, we conducted a manual evaluation. We randomly selected 720 sentences from the general test set and 100 sentences from the discourse test set and had them evaluated separately for adequacy and fluency by two native speakers of Russian. To avoid priming between the fluency and adequacy conditions, the test set was split between the annotators, and no sentence was annotated for adequacy and fluency by the same annotator. To determine the inter-annotator agreement, there are 100 overlapping sentences for two annotators. Table 5 shows inter-annotator agreement results while Table 4 shows the intra-annotator agreement. According to Landis and Koch (1977), all groups of human evaluation results are fair ($\kappa > 0.2$).

The sentences were presented to the annotators in random order along with 3 sentences of preceding context. The sentence to be evaluated was highlighted, and the Russian translations of the three systems (Baseline, DocRepair (+*ParData*) and Transference (+*ParData*)) were displayed next to each other, ordered randomly. In the adequacy condition only, the English source text was also shown. The annotators received instructions according to Table 2 and were told to assign the same rank if two translations were of equal quality. Once the annotation was complete, the rankings were converted into pairwise comparisons. Duplicate assessments from the inter- and intra-annotator sets were counted once if their annotations agreed, and discarded if they disagreed.

Table 3 shows the outcome of pairwise comparisons between the systems, including the number of times the output of one system was preferred over that of the other by the annotator. The results were tested for significance with a sign test. We find the same pattern of results for both test sets. In the *Fluency* evaluation, both monolingual DocRepair and bilingual Transference significantly improve over the Baseline. The comparison between DocRepair and Transference is not significant in this condi-

Adequacy: Please rank the three translations according to how adequately the translation of the last sentence reflects the meaning of the source, given the context.

Fluency: Please rank the three translations according to how fluent the last sentence is, in terms of grammaticality, naturalness and consistency, taking into account the context of the previous sentences.

Table 2: Instructions to human annotators

System A	System B	Preference		
		A	B	Ties
Fluency				
<i>General corpus:</i>				
Baseline	DocRepair	30 < 62	612	($p < 0.005$)
Baseline	Transference	51 < 89	547	($p < 0.005$)
DocRepair	Transference	70 78	542	(n. s.)
<i>Discourse corpus:</i>				
Baseline	DocRepair	12 < 28	138	($p < 0.05$)
Baseline	Transference	15 < 34	120	($p < 0.01$)
DocRepair	Transference	23 25	121	(n. s.)
Adequacy				
<i>General corpus:</i>				
Baseline	DocRepair	24 31	655	(n. s.)
Baseline	Transference	34 < 67	592	($p < 0.005$)
DocRepair	Transference	39 < 66	592	($p < 0.05$)
<i>Discourse corpus:</i>				
Baseline	DocRepair	16 20	140	(n. s.)
Baseline	Transference	9 < 46	117	($p < 0.001$)
DocRepair	Transference	11 < 43	117	($p < 0.001$)

n. s. = not significant
Significance threshold: $p < 0.05$

Table 3: Human evaluation results. Winning systems in pairwise comparisons marked in bold.

tion. In the *Adequacy* evaluation, the comparison between DocRepair and the Baseline is not significant, but Transference significantly outperforms both DocRepair and the Baseline, demonstrating that knowledge of the source is essential for APE to improve the accuracy of the translations.

One of the evaluators provided qualitative comments on 32 pairs of DocRepair and Transference outputs sampled from those sentences for which the two systems were ranked differently in the human evaluation. The comments show that both

<i>Per annotator:</i>			
Annotator 1		91.1%	
Annotator 2		83.9%	
<i>Per dataset:</i>			
Fluency	General	90.0%	
Fluency	Discourse	86.7%	
Adequacy	General	90.0%	
Adequacy	Discourse	78.3%	

Table 4: Intra-annotator agreement of human evaluation

		κ	Pct.
Fluency	General	0.234	5
Fluency	Discourse	0.352	55
Adequacy	General	0.301	27
Adequacy	Discourse	0.471	93

Table 5: Inter-annotator agreement in terms of Cohen’s κ (Cohen, 1960). The last column shows the percentile of our κ value in the context of a series of similar evaluations carried out at WMT 2012–2016 (Bojar et al., 2016, Table 4).

systems tend to produce imperfect output for the same sentences, but the winning system often manages to fix errors partially. Both systems make a wide range of errors in terms of morphology and lexical choice, but the source information permits Transference to correct certain recurring problems more reliably, such as agreement errors, mistranslations of proper names (e.g., Lena as Sarah), or the incorrect use or omission of subjunctive mood in conditional sentences.

7 Related Work

Our work draws on two strands of research: automatic post-editing and context-aware MT.

Automatic post-editing has a long history in MT (Knight and Chander, 1994), with regular shared tasks (Bojar et al., 2015, 2016, 2017). Neural multi-source APE systems as first proposed by Pal et al. (2016) and Junczys-Dowmunt and Grundkiewicz (2016), some of them including source language information (Junczys-Dowmunt and Grundkiewicz, 2017; Chatterjee et al., 2017; Libovický and Helcl, 2017), have come to dominate APE. We take inspiration from the top-performing systems at the WMT19 shared task for architectures and training/decoding tricks (Chatterjee et al., 2019), and make heavy use of synthetic training data (Sennrich et al., 2016a; Junczys-Dowmunt and Grundkiewicz, 2016; Freitag et al., 2019).

Neural context-aware MT can be achieved by integrating context into the main translation model (Jean et al., 2017; Tiedemann and Scherrer, 2017; Bawden et al., 2018, inter alia). Two-stage models with a sentence-level first pass and document-level second pass have been explored for scenarios with asymmetric training data. Voita et al. (2019b) introduces a two-pass model where, unlike in APE, the second-pass is tightly integrated with the first-pass model, reusing its hidden representations. Apart

from Voita et al. (2019a), the model closest to ours is by Junczys-Dowmunt (2019), who explored document-level APE, but only manually evaluated its efficacy as part of a large model ensemble.

8 Conclusion

Our human evaluation shows that monolingual APE oriented towards consistency beyond the sentence level improves fluency, but not adequacy, while multi-source APE with source context improves both adequacy and fluency. This shortcoming of monolingual APE in terms of adequacy was not easily visible with a consistency-focused automatic evaluation, highlighting the need for human evaluation to avoid such blind spots and reinforcing earlier findings about the inadequacy of automatic evaluation methods for discourse-level MT (Guilou and Hardmeier, 2018).

Clearly, a two-stage process with sentence-level translation and multi-sentence APE is a viable approach in asymmetric data settings with little document-level parallel data. However, we still required some actual document-level parallel data, and were unable to match the success of monolingual repair when using only synthetic data. Exploring the data requirements of document-level APE, and devising ways to reduce them, are worth further study.

Acknowledgments

Chaojun Wang was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) fellowship grant EP/S001271/1 (MTStretch). Christian Hardmeier was supported by the Swedish Research Council under grant 2017-930. This project has received funding from the European Union’s Horizon 2020 research and innovation programme (ELITR, grant agreement no 825460), and the Royal Society (NAFR1\180122).

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating Discourse Phenomena in Neural Machine Translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313, New Orleans, USA. Association for Computational Linguistics.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.
- Rajen Chatterjee, M. Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017. Multi-source neural automatic post-editing: FBK’s participation in the WMT 2017 APE shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 630–638, Copenhagen, Denmark. Association for Computational Linguistics.
- Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the WMT 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28, Florence, Italy. Association for Computational Linguistics.
- Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.
- Markus Freitag, Isaac Caswell, and Scott Roy. 2019. Ape at scale and its implications on mt evaluation biases. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 34–44, Florence, Italy. Association for Computational Linguistics.
- Liane Guillou and Christian Hardmeier. 2018. Automatic reference-based evaluation of pronoun translation misses the point. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4797–4802, Brussels, Belgium. Association for Computational Linguistics.
- Christian Hardmeier. 2014. *Discourse in Statistical Machine Translation*, volume 15 of *Studia Linguistica Upsaliensia*. Acta Universitatis Upsaliensis, Uppsala.
- Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does Neural Machine Translation Benefit from Larger Context? In *arXiv:1704.05135*. ArXiv: 1704.05135.
- Marcin Junczys-Dowmunt. 2018. Microsoft’s submission to the WMT2018 news translation task: How I learned to stop worrying and love the data. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 425–430, Belgium, Brussels. Association for Computational Linguistics.
- Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. Association for Computational Linguistics.
- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 751–758, Berlin, Germany. Association for Computational Linguistics.

- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. An exploration of neural sequence-to-sequence architectures for automatic post-editing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 120–129, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Kevin Knight and Ishwar Chander. 1994. Automated postediting of documents. In *AAAI*, volume 94, pages 779–784.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.
- J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.
- Jindřich Libovický and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 196–202, Vancouver, Canada. Association for Computational Linguistics.
- Chin-Yew Lin and Franz Josef Och. 2004. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 501–507, Geneva, Switzerland. COLING.
- Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. OpenSubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- António V. Lopes, M. Amin Farajian, Gonçalo M. Correia, Jonay Trénous, and André F. T. Martins. 2019. Unbabel’s submission to the WMT2019 APE shared task: BERT-based encoder-decoder for automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 118–123, Florence, Italy. Association for Computational Linguistics.
- Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, Brussels, Belgium. Association for Computational Linguistics.
- Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, and Josef van Genabith. 2016. A neural network based approach to automatic post-editing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 281–286, Berlin, Germany. Association for Computational Linguistics.
- Santanu Pal, Hongfei Xu, Nico Herbig, Antonio Krüger, and Josef van Genabith. 2019. USAAR-DFKI – the transference architecture for English–German automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 124–131, Florence, Italy. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Yves Scherrer, Jörg Tiedemann, and Sharid Loáigiga. 2019. Analysing concatenation approaches to document-level NMT in two different domains. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 51–61, Hong Kong, China. Association for Computational Linguistics.
- Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Lüubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nădejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks.

In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Jörg Tiedemann and Yves Scherrer. 2017. Neural Machine Translation with Extended Context. In *Proceedings of the Third Workshop on Discourse in Machine Translation, DISCOMT'17*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 877–886, Hong Kong, China. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019b. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212, Florence, Italy. Association for Computational Linguistics.

A Appendix

A.1 Hyperparameter Search and Validation Performance

The following hyperparameters were manually tuned:

- The percentage of *ParData* mixed with the synthetic training data. of Transference.
- The conservativeness penalty.
- The decision whether to add the conservativeness penalty to the probability estimates or to the logits of the model.

The tuning bounds are shown in Table 7 in curly braces for each tuned hyperparameter. After 18 hyperparameter search trials, the best-performing models were selected considering both BLEU score on the general validation set and the accuracy on the contrastive validation sets. The validation results are shown in Table 6, and the hyperparameter configurations in Table 7.

Model	Deixis	Lex.c.	CE.loss	BLEU
DocRepair	89.0	68.0	58.2	32.01
DocRepair (+ParData)	88.8	68.8	56.3	31.63
Transference	86.0	62.2	61.0	30.37
Transference (+ParData)	85.4	64.8	50.7	31.99

Table 6: Validation performance of tested systems (CE represents Cross Entropy).

A.2 Training Time and Model Size

The two sentence-level baselines and the DocRepair model have approximately 72 million parameters each. The baseline systems are trained for around 72 hours each on a GeForce GTX 1080 Ti GPU. DocRepair and DocRepair (+*ParData*) are trained for approximately 216 hours on four TITAN X (Pascal) GPUs and 192 hours on a GeForce RTX 2080 Ti GPU, respectively.

The Transference model has around 119 million parameters. Transference and Transference (+*ParData*) were trained for around 192 and 288 hours, respectively, on three GeForce GTX 1080 Ti GPUs.

	DocRepair	Transference	Tuning bounds
Common hyperparameters			
Embedding layer size		512	
Hidden state size		512	
Tied encoder/decoder embeddings	yes	no	
Tie decoder embeddings		yes	
Loss function		per-token cross-entropy	
Label smoothing		0.1	
Optimizer		Adam	
Learning schedule		Transformer	
Warmup steps		8000	
Gradient clipping threshold		1.0	
Maximum sequence length		500	
Token batch size		15000	
Length normalization alpha		0.6	
Encoder depth		6	
Decoder depth		6	
Feed forward num hidden		2048	
Number of attention heads		8	
Embedding dropout		0.1	
Residual dropout		0.1	
ReLU dropout		0.1	
Attention weights dropout		0.1	
Beam size		4	
Percentage of ParData in training		0.3	{0.2, 0.3, 0.4}
Transference-specific hyperparameters			
Tied second encoder/decoder embeddings		yes	
Second encoder depth		6	
Conservativeness penalty	(0.2, probability)		{0.1, 0.2, 0.3} × {probability, logit}

Table 7: Hyperparameter configurations for best-performing DocRepair and Transference models, and hyperparameter tuning bounds.

Chinese Character Decomposition for Neural MT with Multi-Word Expressions

Lifeng Han¹, Gareth J. F. Jones¹, Alan F. Smeaton² and Paolo Bolzoni

¹ ADAPT Research Centre

² Insight Centre for Data Analytics

School of Computing, Dublin City University, Dublin, Ireland

lifeng.han@adaptcentre.ie, paolo.bolzoni.brown@gmail.com

Abstract

Chinese character decomposition has been used as a feature to enhance Machine Translation (MT) models, combining radicals into character and word level models. Recent work has investigated ideograph or stroke level embedding. However, questions remain about the different decomposition levels of Chinese character representations, radical and strokes, best suited for MT. To investigate the impact of Chinese decomposition embedding in detail, i.e., radical, stroke, and intermediate levels, and how well these decompositions represent the meaning of the original character sequences, we carry out analysis with both automated and human evaluation of MT. Furthermore, we investigate if the combination of decomposed Multiword Expressions (MWEs) can enhance model learning. MWE integration into MT has seen more than a decade of exploration. However, decomposed MWEs has not previously been explored.

1 Introduction

Neural Machine Translation (NMT) (Cho et al., 2014; Johnson et al., 2016; Vaswani et al., 2017; Lample and Conneau, 2019) has recently replaced Statistical Machine Translation (SMT) (Brown et al., 1993; Och and Ney, 2003; Chiang, 2005; Koehn, 2010) as the state-of-the-art for Machine Translation (MT). However, research questions still remain, such as how to deal with *out-of-vocabulary* (OOV) words, how best to integrate *linguistic knowledge* and how best to correctly translate *multi-word expressions* (MWEs) (Sag et al., 2002; Moreau et al., 2018; Han et al., 2020a). For OOV word translation for European languages, substantial improvements have been

made in terms of rare and unseen words by incorporating sub-word knowledge using Byte Pair Encoding (BPE) (Sennrich et al., 2016). However, such methods cannot be directly applied to Chinese, Japanese and other ideographic languages.

Integrating sub-character level information, such as Chinese ideograph and radicals as learning knowledge has been used to enhance features in NMT systems (Han and Kuang, 2018; Zhang and Matsumoto, 2018; Zhang and Komachi, 2018). Han and Kuang (2018), for example, explain that the meaning of some unseen or low frequency Chinese characters can be estimated and translated using *radicals* decomposed from the Chinese characters, as long as the learning model can acquire knowledge of these radicals within the training corpus.

Chinese characters often include two pieces of information, with *semantics* encoded within radicals and a *phonetic* part. The phonetic part is related to the pronunciation of the overall character, either the same or similar. For instance, Chinese characters with this two-stroke radical, 刂 (tí dāo páng), ordinarily relate to *knife* in meaning, such as the Chinese character 劍 (jiàn, *sword*) and multi-character expression 鋒利 (fēnglì, *sharp*). The radical 刂 (tí dāo páng) preserves the meaning of knife because it is a variation of a drawing of a knife evolving from the original bronze inscription (Fig. 4 in Appendices).

Not only can the radical part of a character be decomposed into smaller fragments of strokes but the phonetic part can also be decomposed. Thus there are often several levels of decomposition that can be applied to Chinese characters by combining different levels of decomposition of each part of the Chinese character. As one example, Figure 1 shows the three decomposition levels from our model and the full stroke form of the above mentioned characters 劍(jiàn) and 鋒(fēng). To date, little work has been carried out to investigate

the full potential of these alternative levels of decomposition of Chinese characters for the purpose of Machine Translation (MT).

In this work, we investigate Chinese character decomposition, and another area related to Chinese characters, namely Chinese MWEs. We firstly investigate translation at increasing levels of decomposition of Chinese characters using underlying radicals, as well as the additional Chinese character strokes (corresponding to ever-smaller units), breaking down characters into component parts as this is likely to reduce the number of unknown words. Then, in order to better deal with MWEs which have a common occurrence in general contexts (Sag et al., 2002), and working in the opposite direction in terms of meaning representation, we investigate translating larger units of Chinese text, with the aim of restricting translation of larger groups of Chinese characters that should be translated together as one unit. In addition to investigating the effects of decomposing characters we simultaneously apply methods of incorporating MWEs into translation. MWEs can appear in Chinese in a range of ways, such as fixed (or semi-fixed) expressions, metaphor, idiomatic phrases, and institutional, personal or location names, amongst others.

In summary, in this paper, we investigate: (i) the degree to which Chinese radical and stroke sequences represent the original word and character sequences that they are composed of; (ii) the difference in performance achieved by each decomposition level; (iii) the effect of radical and stroke representations in MWEs for MT. Furthermore, we offer:

- an open-source suite of Chinese character decomposition extraction tools;
- a Chinese \Leftrightarrow English MWE corpus where Chinese characters have been decomposed

available at [radical4mt](https://github.com/poethan/MWE4MT)¹.

The rest of this paper is organized as follows: Section 2 provides details of related work in character and radical related MT; Sections 3 and 4 introduce our Chinese decomposition procedure into radical and strokes, and our experimental design; Section 5 provides details of our evaluations from both automatic and human perspectives; Section 6 describes conclusions and plans for future work.

¹<https://github.com/poethan/MWE4MT>

2 Related Work

Chinese character decomposition has been explored recently for MT. For instance, Han and Kuang (2018) and Zhang and Matsumoto (2018), considered radical embeddings as additional features for Chinese \rightarrow English and Japanese \Leftrightarrow Chinese NMT. Han and Kuang (2018) tested a range of encoding models including word+character, word+radical, and word+character+radical. This final setting with word+character+radical achieved the best performance on a standard NIST² MT evaluation data set for Chinese \rightarrow English. Furthermore, Zhang and Matsumoto (2018) applied radical embeddings as additional features to character level LSTM-based NMT on Japanese \rightarrow Chinese translation. None of the aforementioned work has however investigated the performance of decomposed character sequences and the effects of varied decomposition degrees in combination with MWEs. Subsequently, Zhang and Komachi (2018) developed bidirectional English \Leftrightarrow Japanese, English \Leftrightarrow Chinese and Chinese \Leftrightarrow Japanese NMT with word, character, ideograph (the phonetics and semantics parts of characters are separated) and stroke levels, with experiments showing that the *ideograph* level was best for ZH \rightarrow EN MT, while the stroke level was best for JP \rightarrow EN MT. Although their ideograph and stroke level setting replaced the original character and word sequences, there was no investigation of *intermediate decomposition* performance, and they only used BLEU score for automated evaluation with no human assessment involved. This gives us inspiration to explore the performance of intermediate level embedding between ideograph and strokes for the MT task.

3 Chinese Character Decomposition

In this section, we introduce a character decomposition approach and the extraction tools which we apply in this work (code will be publicly available). We utilize the open source IDS dictionary³ which was derived from the CHISE (CHaracter Information Service Environment) project⁴. It is comprised of 88,940 Chinese characters from CJK (Chinese, Japanese, Korean script) Unified

²<https://www.nist.gov/programs-projects/machine-translation>

³<https://github.com/cjkvi/cjkvi-ids>

⁴<http://www.chise.org/>

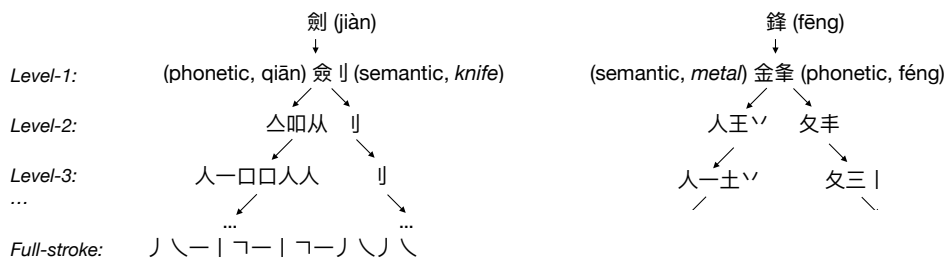


Figure 1: Examples of the decomposition process.

Ideographs and the corresponding decomposition sequences of each character. Most characters are decomposed as a single sequence, but characters can have up to four possible decomposed representations. The reason for this is that the characters can come from different resources, such as Chinese Hanzi (G, H, T for Mainland, Hong Kong, and Taiwan), Japanese Kanji (J), Korean Hanja (K), and Vietnamese ChuNom (V), etc.⁵ Even though they have the same root of Hanzi, the historical development of languages and writing systems in different territories has resulted in certain degrees of variation in their appearance and stroke order. For instance, (且, qiě) vs (目, mù) from the second character example in Figure 2.

Figure 2 shows example characters that have two different decomposition sequences. In our experiments, when there is more than one decomposed representation of a given character, we choose the Chinese mainland decomposition standard (G) for the model, since the corpora we use correspond best to simplified Chinese as used in mainland China. The examples in Figure 2 also show the general construction and corresponding decomposition styles of Chinese characters, such as *left-right*, *up-down*, *inside-outside*, and *embedded* amongst others. To obtain a decomposition level L representation of Chinese character α , we go through the IDS file L times. Each time, we search the IDS file character list to match the newly generated smaller sized characters and replace them with decomposed representation recursively.

4 NMT Experiments

We test the various levels of decomposed Chinese and Chinese MWEs using publicly available data from the WMT-2018 shared tasks Chinese to English

⁵Universal Coded Character Set (10646:2017) standards.iso.org/ittf/PubliclyAvailableStandards

Character	Decomposition	Decomposition
丽 (lì)	日一田田冫、田冫、[G]	日田一田冫、日一田冫、[T]
具 (jù)	日田且一八[GTKV]	日目一八[J]
函 (hán)	冫口田了日彡八[GTV]	冫口田彡日彡八[JK]
勇 (yǒng)	日甬力[GTV]	日田彡田力[JK]

Character construction: 日: up-down, 田: left-right, 冫: inside-outside, 田: embedded

Figure 2: Character examples from IDS dictionary; the grey parts of decomposition graphs represent the construction structure of the character.

English, using the preprocessed (word segmented) data as training data (Bojar et al., 2018). We preserve the original word boundaries in decomposition sequences. To get better generalizability of our decomposition model, we use a large size training set, the first 5 million parallel sentences for training across all learning steps. The corpora “newsdev2017” used for development and “newstest2017” for testing are from the WMT-2017 MT shared task (Bojar et al., 2017). These include 2002 and 2001 parallel Chinese \leftrightarrow English respectively. We use the THUMT (Zhang et al., 2017) toolkit which is an implementation of several attention-based Transformer architectures (Vaswani et al., 2017) for NMT and set up the encoder-decoder as 7+7 layers. Batch size is set as 6250. For sub-word encoding BPE technology, we use 32K BPE operations that are learned from the bilingual training set. We use Google’s Colab platform to run our experiments⁶. We call our baseline model using character sequences (with word boundary) the *character sequence model*. For MWE integrated models, we apply the same bilingual MWE extraction pipeline from our previous work (Han et al., 2020b), similar to (Rikters and

⁶<https://colab.research.google.com>

	20k	100k	120k	140k	160k	180k
baseline	18.39	21.56	21.45	21.31	21.29	21.42
base+MWE	18.49	21.39	21.67	21.83	21.42	21.86
RXD3	16.48	20.75	20.73	20.93	20.98	21.14
RXD3+MWE	17.82	21.36	21.50	21.31	21.42	21.47
RXD2	11.84	13.26	12.88	13.02	13.38	12.86
RXD1/ideograph	15.52	20.67	20.61	21.26	20.76	21.00

Figure 3: Chinese→English BLEU scores for increasing learning steps; RXD1/2/3 represents the decomposition level of Chinese characters. RXD1 indicates *ideograph* from (Zhang and Komachi, 2018)

Bojar, 2017), which is an automated pre-defined PoS pattern-based extraction procedure with filtering threshold set to 0.85 to remove lower quality translation pairs. We integrate these extracted bilingual MWEs back into the training set to investigate if they help the MT learning. In the decomposed models, we replace the original Chinese character sequences from the corpus with decomposed character-piece sequence inputs for training, development and testing (keeping the original word boundary).

5 Evaluation

In order to assess the performance of each model employing a different meaning representation in terms of decomposition and MWEs, we carried out both automatic evaluation using BLEU (Papineni et al., 2002) in Fig. 3, and human evaluation (Direct Assessment) of the outputs of the system. Since decomposition level 3 yields generally higher scores than the other two levels, we also applied decomposition of MWEs to level 3 and concatenated the bilingual glossaries to the training.

We used the models with the most learning steps, 180K, and run human evaluation on the Amazon Mechanical Turk crowd-sourcing platform,⁷ including the strict quality control measures of Graham et al. (2016). Direct Assessment scores for systems were calculated as in Graham et al. (2019) by firstly computing an average score per translation before calculating the overall average for a system from its average scores for translations. Significance tests in the form of Wilcoxon Rank-Sum test are then applied to score distributions of the latter to identify systems that significantly outperform other systems in the human evaluation.

⁷<https://www.mturk.com>

Results of the Direct Assessment human evaluation are shown in Table 1 where similarly performing systems are clustered together (denoted by horizontal lines in the table). Systems in a given lower ranked cluster are significantly outperformed by all systems in a higher ranked cluster. Amongst the six models included in the human evaluation, the first five form a cluster with very similar performance according to human assessors, including the baseline, MWE, RXD1, RXD3MWE, and RXD3 which do not outperform each other with any significance. RXD2, on the other hand, is far behind the other models in terms of performance according to human judges (also the automated BLEU score) performing significantly worse than all other runs (at $p < 0.05$). As the tradition of WMT shared task workshop, we cluster the first five models into one group, while the RXD2 into a second group. Furthermore, human evaluation results in Table 1 show that the top five models all achieve high performance on-par with state-of-the-art in Chinese to English MT.

We also discovered that the decomposed models generated fewer system parameters for the neural nets to learn, which potentially reduces computational complexity. For instance, the total trainable variable size of the character sequence baseline model is 89,456,896, while this number decreased to 80,288,000 and 80,591,104 respectively for the RXD3 and RXD2 models (a 10.25% drop for RXD3). As mentioned by Goodfellow et al. (2016), in NLP tasks the total number of possible words is so large that the word sequence models have to operate on an extremely high-dimensional and sparse discrete space. The decomposition model reduced the overall size of possible tokens for the model to learn, which is more space efficient.

For the automatic and human evaluation results, where decomposition level 2 achieved a surprisingly lower score than the other levels, error analysis revealed an important insight. While level 1 decomposition encoded the original character sequences into radical representations, and this typically contains semantic and phonetic parts of the character, and level 3 gives a deeper decomposition of the character such as the stroke level pieces with sequence order. In contrast, however, level-2 decomposition appears to introduce some intermediate characters that mislead model learning. These intermediate level characters are usu-

Ave.	Ave. z	n	N	
raw				
73.2	0.161	1,232	1,639	BASE
71.6	0.125	1,262	1,659	MWE
71.6	0.113	1,257	1,672	RXD1
71.3	0.109	1,214	1,593	RXD3MWE
70.2	0.073	1,260	1,626	RXD3
53.9	-0.533	1,227	1,620	RXD2

Table 1: Human evaluation results for systems using Direct Assessment, where Ave. raw = the average score for translations calculated from raw Direct Assessment scores for translations, Ave. z = the average score for translations after score standardization per human assessor mean and standard deviation score, n is the number of distinct translations included in the human evaluation (the sample size used in significance testing), N is the number of human assessments (including repeat assessment).

ally constructed from fewer strokes than the original root character, but can be decomposed from it. As in Figure 1, from decomposition level 2, we get new characters 从 (cóng) and 王 (wáng) respectively from 劍 (Jiàn, *sword*) and 鋒 (fēng, *edge/sharp point*), but they have no direct meaning from their father characters, instead meaning “from” and “king” respectively. In summary, decomposition level-2 tends to generate some intermediate characters that do not preserve the meaning of the original root character’s radical, nor those of the strokes, but rather smaller sized independent characters with fewer strokes that result in other meanings.

6 Conclusions and Future Work

In this work, we examined varying degrees of Chinese character decomposition and their effect on Chinese to English NMT with attention architecture. To the best of our knowledge, this is the first work on detailed decomposition level of Chinese characters for NMT, and decomposition representation for MWEs. We conducted experiments for decomposition levels 1 to 3; we had a look at level 4 decomposition and it appears similar to level 3 sequences. We publish our extraction toolkit free for academic usage. We conducted automated evaluation with the BLEU metric, and

crowd sourced human evaluation with the direct assessment (DA) methodology. Our conclusion is that the Chinese character decomposition levels 1 and 3 can be used to represent or replace the original character sequence in an MT task, and that this achieves similar performance to the original character sequence model in our NMT setting. However, decomposition level 2 is not suitable to represent the original character sequence in meaning at least for MT. We leave it to future work to explore the performance of different decomposition levels in other NLP tasks.

Another finding from our experiments is that while adding bilingual MWE terms can both increase character and decomposed level MT score according to the automatic metric BLEU, the human evaluation shows no statistical significance between them. Significance testing using automated evaluation metrics will be carried out in our future work, such as METEOR (Banerjee and Lavie, 2005), and LEPOR (Han et al., 2012; Han, 2014), in addition to BLEU.

We will consider different MWE integration methods in future and reduce the training set to investigate the differences in low-resource scenarios (5 million sentence pairs for training set were used in this work). We will also sample a set of the testing results and conduct a human analysis regarding the MWE translation accuracy from different representation models. We will further investigate different strategies of *combining* several level of decompositions together and their corresponding performances in semantic representation, such as MT task. The IDS file we applied to this work limited the performance of full stroke level capability, and we will look for alternative methods to achieve full-stroke level character sequence extraction for NLP tasks investigation.

Acknowledgments

We thank Yvette Graham for helping with human evaluation, Eoin Brophy for helps with Colab, and the anonymous reviewers for their thorough reviews and insightful feedback. The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund. The input of Alan Smeaton is part-funded by Science Foundation Ireland under grant number SFI/12/RC/2289 (Insight Centre).

References

- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL*.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.
- Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (wmt18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 272–307, Belgium, Brussels. Association for Computational Linguistics.
- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.
- David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan. Association for Computational Linguistics.
- KyungHyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2016. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, FirstView:1–28.
- Yvette Graham, Barry Haddow, and Philipp Koehn. 2019. Translationese in machine translation evaluation. *CoRR*, abs/1906.09833.
- Lifeng Han. 2014. *LEPOR: An Augmented Machine Translation Evaluation Metric*. University of Macau.
- Lifeng Han, Gareth Jones, and Alan Smeaton. 2020a. AlphaMWE: Construction of multilingual parallel corpora with MWE annotations. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 44–57, online. Association for Computational Linguistics.
- Lifeng Han, Gareth Jones, and Alan Smeaton. 2020b. MultiMWE: Building a multi-lingual multi-word expression (MWE) parallel corpora. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2970–2979, Marseille, France. European Language Resources Association.
- Lifeng Han and Shaohui Kuang. 2018. Incorporating chinese radicals into neural machine translation: Deeper than character level. In *Proceedings of ESSLLI-2018 Student Session*, pages 54–65. Association for Logic, Language and Information (FoLLI).
- Lifeng Han, Derek F. Wong, and Lidia S. Chao. 2012. Lepor: A robust evaluation metric for machine translation with augmented factors. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, page 441–450. Association for Computational Linguistics.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *CoRR*, abs/1611.04558.
- Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.
- Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.
- Erwan Moreau, Ashjan Alsulaimani, Alfredo Maldonado, Lifeng Han, Carl Vogel, and Koel Dutta Chowdhury. 2018. Semantic reranking of CRF label sequences for verbal multiword expression identification. In *Multiword expressions at length and in depth: Extended papers from the MWE 2017 workshop*, pages 177 – 207. Language Science Press.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.
- Matīss Rikters and Ondřej Bojar. 2017. Paying Attention to Multi-Word Expressions in Neural Machine Translation. In *Proceedings of the 16th Machine Translation Summit (MT Summit 2017)*, Nagoya, Japan.
- Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword

expressions: A pain in the neck for nlp. In *Computational Linguistics and Intelligent Text Processing*, pages 1–15, Berlin, Heidelberg, Springer Berlin Heidelberg.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Conference on Neural Information Processing System*, pages 6000–6010.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huan-Bo Luan, and Yang Liu. 2017. Thumt: An open source toolkit for neural machine translation. *ArXiv*, abs/1706.06415.

Jinyi Zhang and Tadahiro Matsumoto. 2018. Improving character-level japanese-chinese neural machine translation with radicals as an additional input feature. *CoRR*, abs/1805.02937.

Longtu Zhang and Mamoru Komachi. 2018. Neural machine translation of logographic language using sub-character level information. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 17–25, Brussels, Belgium. Association for Computational Linguistics.

Appendices

Appendix A: Chinese Character Knowledge

Figure 4 demonstrates the meaning preservation root of Chinese radicals, where the evolution of the Chinese character 刀 (Dāo), meaning *knife*, evolved from bronze inscription form to contemporary character and radical form, 刂 (named as: tí dāo páng).

NMT for Asian languages has included translation at the level of phrase, word, and character sequences (see Figure 5).

Appendix B: More Details of Evaluation

The evaluation scores of character sequence baseline NMT, character decomposed NMT and MWE-NMT according to the BLEU metric are presented in Fig. 3. The RXD1 model, decomposition level 1, is the *ideograph* model Zhang and Komachi (2018) used for their experiments where the phonetics (声旁 shēng páng) and semantics (形旁 xíng páng) parts of character are separated initially.

From the automated evaluation results, we see that decomposition model RXD3 has very close BLEU scores to the baseline character sequence (both with word boundary) model. This is very interesting since the level 3 Chinese decomposition is typically impossible (or too difficult) for even native language human speakers to read and understand. Furthermore, by adding the decomposed MWEs back into the learning corpus, “rxd3+MWE” (RXD3MWE) yields higher BLEU scores in some learning steps than the baseline model. To gain further insight, we provide the learning curve with the learning steps and corresponding automated-scores in Figure 6.

The BLEU score increasing ratio in decomposed models (from RXD3 to RXD3MWE) is larger than the ratio in original character sequence models (from BASE to BASEMWE) by adding MWEs in general. Furthermore, the increase in performance is very consistent by adding MWEs from the decomposed model, compared to the conventional character sequence model. For instance, the performance has a surprisingly drop at 100K learning steps for BASEMWE.

Appendix C: Looking into MT Examples

From the learning curves in Fig. 6, we suggest that with 5 million training sentences and 7+7 layers of encoder-decoder neural nets, the Transformer model becomes too flat in its learning rate curve with 100K learning steps, and this applies to both original character sequence model and decomposition models.

In light of this, we look at the MT outputs from head sentences of testing file at 100K learning steps models, and provide some insight into errors made by each model. Even though the automated BLEU metric gives the baseline model a higher score 21.56 than the RXD3 model (20.75) the translation of some Chinese MWE terms is better with the RXD3 model. For instance, in Figure 7, the Chinese MWE 商场 (Shāngchǎng) in the first sentence is correctly translated as *mall* by RXD3 model but translated as *shop* by the baseline character sequence model; the MWE 楼梯间 (lóutījiān) in the second sentence is correctly translated as *stairwell* by the RXD3 model while translated as *stairs* by baseline. Furthermore, the MWE 近日 (Jìnrì) meaning *recently* is totally missed out by the original character sequence model, which results in a misleading am-

Chinese radical 刂 (Dāo, knife) evolution from Pictogram to Regular script					
商 Shang Dynasty (1600-1046BC)		西周 Western-Zhou Dynasty (1045-771BC)	戰國 Warring States period (476-221BC)	漢 Han Dynasty (202BC-220)	東漢 Eastern Han (from 57AD on)
Bronze inscriptions	Oracle bone script	Bronze Inscription	Silk	篆 (on Seal)	Regular script

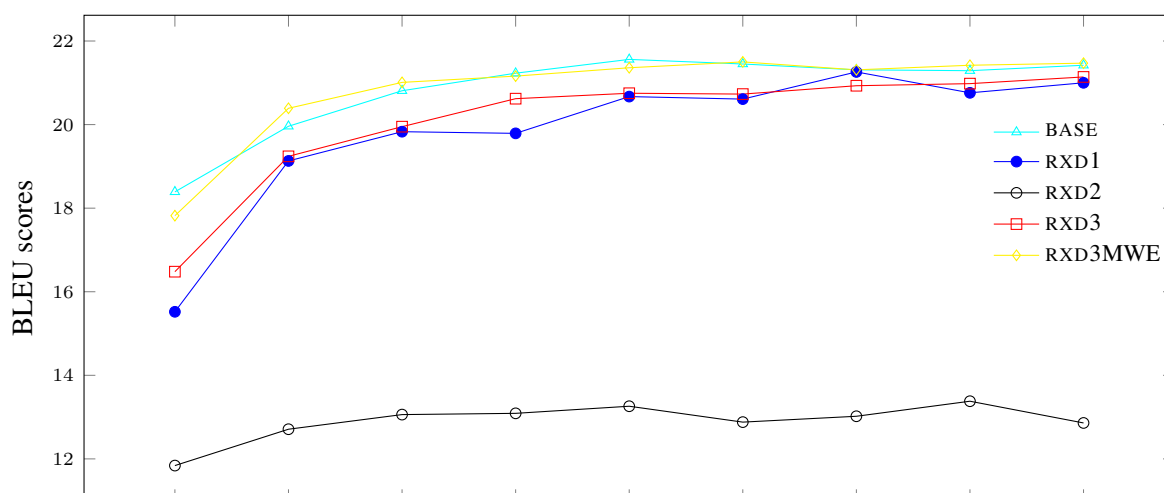
Figure 4: Example Chinese radical, 刂 (Dāo), where the character evolved from leftmost pictogram to present day regular script (rightmost) containing only two strokes. The two strokes are called as 豎 (Shù, vertical) + 豎 (Shù gōu, vertical with hook). The corresponding character representation is 刀 (Dāo).

Word level	28 / 歲 / 廚師 / 被 / 發現 / 死 / 於 / 舊金山 / 一家 / 商場
Character	28 歲 廚 師 被 發 現 死 於 舊 金 山 一 家 商 場
Pronunciation	èr shí bā Suì chū shī bèi fā xiàn sǐ yú jiù jīn shān yī jiā shāng chǎng
Radical	28 止 戌 少 戶 對 白 巾 襪 皮 殳 王 見 歹 匕 方 令 萑 白 人 王 冫 山 一 冫 冫 冫 冫 冫 冫 冫 冫 冫
English Ref.	28-Year-Old Chef Found Dead at San Francisco Mall

Figure 5: Example of Chinese word to character level changes for MT. Pronunciation is Mandarin in Pinyin. The English reference here is taken from the corpus we used for our experiments.

ambiguous translation of an even larger content, i.e., did the chief moved to San Francisco (SF) *recently* or *this week*. We will not get this clearly from the character base sequence model, however, the MWE 近日 (Jìnrì) is correctly translated by the RXD3 model and the overall meaning of the sentence is clear that the chef moved to SF *recently* and was found dead *this week*.

We also attach the translations of these two sentences by four other models. With regard to the first sentence MWEs, all the four models translate San Francisco mall correctly as REF and RXD3 beating BASE model. In terms of the second sentence MWEs, BASEMWE and RXD2 drop out the MWE 近日 (Jìnrì, *recently*) as BASE model, and all the four models drop out the translation of MWE 楼梯间 (lóutījiān, *stairwell*).



src	28岁厨师被发现死于旧金山一家商场 近日刚搬至旧金山的一位28岁厨师本周被发现死于当地一家商场的楼梯间。
ref	28 @-@ year @-@ Old Chef Found Dead at San Francisco mall a 28 @-@ year @-@ old chef who had recently moved to San Francisco was found dead in the stairwell of a local mall this week .
rx3	the 28 @-@ year @-@ old chef was found dead at a San Francisco mall a 28 @-@ year @-@ old chef who recently moved to San Francisco has been found dead on a stairwell in a local mall this week .
base	the 28 @-@ year @-@ old chef was found dead in a shop in San Francisco a 28 @-@ year @-@ old chef who has moved to San Francisco this week was found dead on the stairs of a local mall .
base MWE	28 @-@ year @-@ old chef was found dead at a San Francisco mall a 28 @-@ year @-@ old chef who recently moved to San Francisco was found dead this week at a local mall .
rx3 MWE	28 @-@ year @-@ old chef was found dead at a San Francisco mall a 28 @-@ year @-@ old chef recently moved to San Francisco was found dead this week at a local mall .
rx1	the 28 @-@ year @-@ old chef was found dead at a San Francisco mall a 28 @-@ year @-@ old chef recently moved to San Francisco was found dead in a local shopping mall this week .
rx2	the 28 @-@ year @-@ old chef was found dead in a San Francisco mall a 28 @-@ year @-@ old San Francisco chef was found dead in a local mall this week .

Figure 7: Samples of the English MT output at 100K learning steps: RXD1, RXD2 and RXD3 are the Chinese decomposition with level 1 to 3, BASE is the character sequence model, BASEMWE and RXD3MWE are character sequence model with MWEs and decomposition level 3 model with decomposed MWEs, and src/ref represents source/reference.

Grapheme-Based Cross-Language Forced Alignment: Results with Uralic Languages

Juho Leinonen
Aalto University
juho.leinonen
@aalto.fi

Sami Virpioja
Helsinki University
sami.virpioja
@helsinki.fi

Mikko Kurimo
Aalto University
mikko.kurimo
@aalto.fi

Abstract

Forced alignment is an effective process to speed up linguistic research. However, most forced aligners are language-dependent, and under-resourced languages rarely have enough resources to train an acoustic model for an aligner. We present a new Finnish grapheme-based forced aligner and demonstrate its performance by aligning multiple Uralic languages and English as an unrelated language. We show that even a simple non-expert created grapheme-to-phoneme mapping can result in useful word alignments.

1 Introduction

Matching speech signal and its orthographic transcription is a necessary first step for many research questions in linguistics (Yuan et al., 2018; Olsen et al., 2017; DiCanio et al., 2013). For well-resourced languages, manually aligned corpora exist, providing an easy starting point for linguistic research. For under-resourced languages such corpora are rare, and for all languages new corpora are continuously studied. In these situations, the researcher needs to complete this task before any actual research can begin. Forced alignment, i.e., automatically matching text to speech using automatic speech recognition (ASR), is widely used, and tools that can accomplish this automatically exist, such as FAVE (Rosenfelder et al., 2011), Prosodylab-aligner (Gorman et al., 2011), MAUS (Kisler et al., 2017), and Montreal Forced aligner (MFA) (McAuliffe et al., 2017).

If the researcher is studying a language that is supported by an existing tool for forced alignment, learning to use it will be beneficial, since manual segmentation is much more arduous than transcription (Jarifi et al., 2008). However, the effort for this necessary, but often uninteresting step increases tremendously if no suitable model exists.

The reason may be that the target data is out-of-domain of what the acoustic model was trained with, or the target language is under-resourced and there is no model available at all. Some aligning tools do not support retraining models. For others, such as FAVE and Prosodylab, the model has been trained with a known ASR framework, here HTK (Young et al., 2002), and the researcher could use the framework to train their own models. However, at this point it would be more straightforward to use the ASR framework itself. In addition to all of this, the technical knowledge required to train an acoustic model with minimal or difficult data is formidable.

MFA provides ample documentation, and has a user friendly wrapper over Kaldi (Povey et al., 2011), a popular speech recognition framework. It gives users the option to retrain the model to fit their own data, and add new languages. Gonzalez et al. (2018) used MFA to experiment on iterative forced alignment, and how it compared to the traditional linear method. Even though they used a ready-made tool, the effort to try two alignment methods on an under-resourced language was enough to qualify as a research paper on its own right. For a linguist, who might not have technical expertise on ASR, this may be intimidating as the first step.

An alternative solution to the task of training new models is cross-language forced alignment, in which an aligner trained with a different language than the speech and transcriptions to be aligned, is used. In this paper we introduce a new word-level forced alignment tool based on Kaldi. We show that this very simple command line tool can align closely related languages, is robust against speaker variability without any fine-tuning, and can even adequately align linguistically very dissimilar languages. This paper shows the first results for cross-language forced alignment involving Finnish. In addition, using the tool we force-

aligned a Northern Sámi corpus without proper word alignments with very little expert knowledge of the language.

2 Related research

2.1 Forced aligners

In their paper (McAuliffe et al., 2017), the designers of MFA compared their tool to FAVE and Prosodylab. The latter tools are based on mono-phone models, while MFA utilizes triphones, and adds speaker adaptation to the process. A central underlying difference is that, similar to us, MFA uses Kaldi as the speech recognition framework. However, MFA uses Gaussian mixture models (GMM), popular in speech recognition before deep neural networks (DNN), while our tool uses the modern machine learning methods trained with Kaldi’s lattice-free maximum mutual information cost function (Hadian et al., 2018). Another Kaldi-based tool is Gentle¹, which also uses DNNs. Munich AUtomatic Segmentation system (MAUS) is a popular aligner based on its own speech recognition framework, utilizing a statistical expert system of pronunciation.

2.2 Cross-language forced alignment

Forced alignment has also been successfully used across languages, e.g., when the target language does not have enough transcribed data. This task is called cross-language or cross-linguistic forced alignment (CLFA), sometimes untrained forced alignment. Kempton et al. (2011) used their own phonetic distance metric to evaluate the accuracy of three phoneme recognizers on isolated words from under-resourced language, and again in (Kempton, 2017) to a different target language. In another early experiment (DiCanio et al., 2013), tools trained on English were used to align isolated words from Yoloxóchitl Mixtec. Free conversations were aligned in (Kurtic et al., 2012), where authors tested multiple phoneme recognizers on Bosnian Serbo-Croatian.

Most of the tools introduced at the start of this section have also been tried for CLFA. The authors of MAUS experimented a language-independent ‘sampa’ version on a multitude of under-resourced languages by comparing word start and end boundaries (Strunk et al., 2014). Later Jones et al. (2019) compared MAUS’ language-independent and Italian versions for conversational speech in

¹<https://github.com/lowerquality/gentle>

Kriol, finding that the Italian version surpassed the language-independent one.

A unifying method was presented by Tang and Bennett (2019), who combined a larger source language and the target language with MFA to train the aligner. Finally Johnson et al. (2018) reviewed previous CLFA research and experimented on the minimum amount of data necessary for language dependent forced alignment, achieving good results with an hour of transcribed speech.

3 Experiments

We evaluate our Kaldi-based aligner on related and unrelated languages, with a small amount of expert knowledge added to grapheme-to-phoneme mapping. We also experiment on speaker variation. This is the first time either has been done in CLFA literature. The code and tool used in this paper are publicly available.²

3.1 Kaldi pipeline

Our method uses Kaldi to force-align transcribed audio. As is customary in Kaldi when aligning speech with neural networks, we employ 39 dimension Mel-frequency cepstral coefficients (MFCCs) and Cepstral mean and variance normalization (CMVN). Kaldi’s i-vectors are used for speaker adaptation. The original Finnish acoustic model and i-vector extractor are the same as in (Mansikkaniemi et al., 2017). After the feature generation we create a dataset-specific dictionary from all the words in the transcription. The orthography is assumed to be phonetic, so the words in the lexicon are composed of their graphemes, which are mapped to closest Finnish match manually by non-experts. Smit et al. (2021) show that with DNN-based acoustic models, the assumption of phonetic orthography works reasonably well even for a language like English. As a final preparation for alignment Kaldi uses the lexicon, acoustic model and transcripts to create dataset-specific finite state transducers.

3.2 Datasets

We first evaluate the model on Finnish data using manually annotated Finnish read speech from one male speaker (Vainio, 2001; Raitio et al., 2008). We use Pympi (Lubbers and Torreira, 2013-2015)

²<https://github.com/aalto-speech/finnish-forced-alignment>

to prepare the data. Here the grapheme-to-phoneme mapping is one to one due to Finnish being a phonetic language. For experimenting on speaker variability and CLFA, we align nine Estonian speakers with data gathered from the corpus of lecture speeches introduced in (Meister et al., 2012). For each speaker we have little over 15 minutes of speech, much less than the recommended hour by Johnson et al. (2018). We create a rough mapping between Estonian graphemes and Finnish phonemes, which is a straightforward task as the languages are closely related. We also evaluate our model on Northern Sámi, by force-aligning the Giellagas corpus (Kielipankki, 2014-2017). Since there are no accurate word boundaries for the dataset, we use ELAN (Wittenburg et al., 2006) to manually annotate roughly 20 seconds of speech from 11 native speakers to compare to our automatically generated boundaries. The annotations should be considered only approximate, as the recorded speech has poor quality and the annotator did not know the Sámi language. For Northern Sámi, we use the grapheme-to-phoneme mapping introduced by Leinonen (2015). While most of CLFA papers use closely related or otherwise similar languages, we also try to align English speech with our Finnish model using the clean test sets from Librispeech corpus (Panayotov et al., 2015). For the lexicon we map the graphemes e, and y to Finnish i, and a to ä, otherwise assuming one-to-one mapping.

For all datasets, we follow McAuliffe et al. (2017), and compare what percentage of absolute differences in word start and end boundaries are inside the ranges 10, 25, 50 and 100 milliseconds, when comparing the aligner’s results to the gold standard boundaries. Since we do not have manual alignments for the English and Estonian datasets, we align the audio with language-dependent acoustic models and use the predicted boundaries as gold standards. For Estonian this is done with a dockerized Estonian aligner³. The Librispeech datasets were aligned with an acoustic model trained with Kaldi Librispeech recipe⁴. We use the final GMM-based model called tri6b to create the word boundaries. We also experiment with other triphone models trained with the Librispeech recipe, varying in the amounts of training data, and model complexity, to test what improve-

³<https://github.com/alumae/kaldi-align-server>

⁴<https://github.com/kaldi-asr/kaldi/tree/master/egs/librispeech/s5>

ments the advances in triphone models bring, and how well our Finnish model compares to language dependent models. Table 1 summarizes the sizes of studied datasets.

Lang	Dataset	length	tokens
fin	Finnish	1h7m27s	6464
est	al	16m41s	1910
	ao	16m45s	2199
	hv	16m40s	1697
	jp	16m46s	1953
	mk	16m41s	2602
	ms	16m48s	1523
	mj	16m48s	1394
eng	dev-clean	5h23m16s	54402
	test-clean	5h24m12s	52576
	smi	3min19s	384

Table 1: Speech and text data used for evaluations, with initials of the participant names for Estonian data as they were in the corpus.

4 Results

The Finnish alignment results in Table 2 are quite comparable to what McAuliffe et al. (2017) achieved using MFA for the English Buckeye corpus (Pitt et al., 2005). This seems reasonable since both are using Kaldi. The different amounts of smaller boundary errors might be due to audio quality, speaking style or method of annotation. For instance the Finnish dataset was more focused on phoneme labels than word boundaries.

Model	Dataset	<10	<25	<50	<100
Finnish	Finnish	0.21	0.55	0.84	0.98
MFA	Buckeye	0.33	0.68	0.88	0.97

Table 2: Differences in word boundary accuracy between language-dependent forced alignment. MFA results from (McAuliffe et al., 2017) using the English Buckeye corpus.

When analysing the Estonian results in Table 3, they look comparable to Finnish. Aside from the last 100ms range, they are very similar to MFA’s results for Buckeye. And for smaller ranges are actually better than Finnish alignments. This can be due to similarities in how the speech recognizers generally align speech. Speaker variation is small,

Speaker	<10	<25	<50	<100
al	0.32	0.65	0.82	0.90
ao	0.36	0.72	0.89	0.94
hv	0.32	0.64	0.81	0.88
jp	0.37	0.67	0.83	0.90
mk	0.29	0.59	0.77	0.88
ms	0.33	0.64	0.82	0.89
mj	0.38	0.70	0.86	0.92
mr	0.30	0.62	0.84	0.93
th	0.34	0.64	0.81	0.89
Median	0.33	0.64	0.82	0.90
Std	0.027	0.038	0.033	0.02

Table 3: Cross-language forced alignment for Estonian: results of word boundary accuracy for speaker-wise alignments with median and standard deviation.

with standard deviation being 0.02-0.038. Overall, compared to how well MFA aligned English speech, this is a more fat-tailed distribution, with 10% of boundary errors being larger than 100ms.

Dataset	<10	<25	<50	<100
Giellagas	0.12	0.26	0.45	0.62

Table 4: Cross-language forced alignment for Northern Sámi: word boundary accuracy using a part of the Giellagas corpus.

The results for Northern Sámi in Table 4 are not as good as for Estonian, with some of the possible reasons listed in Section 3.2. With closer inspection of the differences between manual and forced alignment, it could be argued that the automatic method is more accurate. It is definitely much faster, being seconds instead of taking hours.

Dataset	<10	<25	<50	<100
dev-clean	0.12	0.30	0.51	0.68
test-clean	0.12	0.30	0.51	0.67

Table 5: Cross-language forced alignment for English: word boundary accuracy using Librispeech datasets.

The results for English in Table 5 are weaker than for any other target language, with the largest 100ms range having the same results as 25ms range for Estonian. While any researcher who needs to align English speech naturally has language-dependent models, this demonstrates the

worst case scenario for CLFA, with multiple wrong assumptions including rough grapheme-to-phoneme mapping, and even using phonetic orthography. If there is very little target speech, using an unrelated source language might be more cost effective than trying to train a new model or manual alignment.

Model	<10	<25	<50	<100
tri1	0.55	0.87	0.97	1.00
tri2b	0.65	0.93	0.98	1.00
tri3b	0.72	0.95	0.99	1.00
tri4b	0.80	0.97	0.99	1.00
tri5b	0.88	0.99	1.00	1.00

Table 6: Librispeech word boundary accuracy with different English HMM-GMM models trained with Librispeech recipe. Dataset is dev-clean, using tri6b as a gold standard.

The authors of MFA hypothesize the effects of using different phone models, speaker adaptive training and other methods in (McAuliffe et al., 2017). Also to give context to the Finnish-English results, we experimented on how simpler ASR models might perform at the task. Table 6 show that improving the basic model underneath does improve the results for the smallest ranges, and that a much simpler language-dependent model is much better than results with cross-language alignment.

5 Future work

Most of the papers in related research use some tool to automatically generate a phoneme-based lexicon for the target language. These lexicons do contain errors, so we have evaluated our results with word boundaries, since the words can be extracted as is from the transcription. However, automatic phoneme mapping would be an interesting next step, and allow better comparison with previous research effort in this multidisciplinary field.

6 Conclusion

We have demonstrated promising results for cross-language forced alignment using Finnish acoustic model for related and unrelated languages. We have shown that its results for Finnish in language-dependent use are comparable to state-of-the-art aligners for English data. In addition, we present promising results with related and unrelated languages. We also showed the effects of speaker

variation in cross-language situations, demonstrating that retraining speaker dependent models is generally not necessary. We share our tool as an easy to use Docker image.

Acknowledgments

We acknowledge the computational resources provided by the Aalto Science-IT project. SV was supported by the FoTran project, funded by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement № 771113).

References

- Christian DiCanio, Hosung Nam, Douglas H Whalen, H Timothy Bunnell, Jonathan D Amith, and Rey Castillo García. 2013. Using automatic alignment to analyze endangered language data: Testing the viability of untrained alignment. *The Journal of the Acoustical Society of America*, 134(3):2235–2246.
- Simon Gonzalez, Catherine Travis, James Grama, Danielle Barth, and Sunkulp Ananthanarayan. 2018. Recursive forced alignment: A test on a minority language. In *Proceedings of the 17th Australasian International Conference on Speech Science and Technology*, volume 145, page 148.
- Kyle Gorman, Jonathan Howell, and Michael Wagner. 2011. Prosodylab-aligner: A tool for forced alignment of laboratory speech. *Canadian Acoustics*, 39(3):192–193.
- Hossein Hadian, Hossein Sameti, Daniel Povey, and Sanjeev Khudanpur. 2018. End-to-end speech recognition using lattice-free mmi. In *Interspeech*, pages 12–16.
- Safaa Jarifi, Dominique Pastor, and Olivier Rosenc. 2008. A fusion approach for automatic speech segmentation of large corpora with application to speech synthesis. *Speech communication*, 50(1):67–80.
- Lisa M Johnson, Marianna Di Paolo, and Adrian Bell. 2018. Forced alignment for understudied language varieties: Testing prosodylab-aligner with tongan data. *Language Documentation & Conservation*, 12:80–123.
- Caroline Jones, Weicong Li, Andre Almeida, and Amit German. 2019. Evaluating cross-linguistic forced alignment of conversational data in north australian kriol, an under-resourced language. *Language Documentation and Conservation*, pages 281–299.
- Timothy Kempton. 2017. Cross-language forced alignment to assist community-based linguistics for low resource languages. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 165–169, Honolulu. Association for Computational Linguistics.
- Timothy Kempton, Roger K Moore, and Thomas Hain. 2011. Cross-language phone recognition when the target language phoneme inventory is not known. In *Twelfth Annual Conference of the International Speech Communication Association*.
- Kielipankki. 2014-2017. Pohjoissaamen näytekorpus. [Http://urn.fi/urn:nbn:fi:lb-201407302](http://urn.fi/urn:nbn:fi:lb-201407302).
- Thomas Kisler, Uwe Reichel, and Florian Schiel. 2017. Multilingual processing of speech via web services. *Computer Speech & Language*, 45:326 – 347.
- Emina Kurtic, Bill Wells, Guy J Brown, Timothy Kempton, and Ahmet Aker. 2012. A corpus of spontaneous multi-party conversation in bosnian serbo-croatian and british english. In *LREC*, pages 1323–1327. Citeseer.
- Juho Leinonen. 2015. Automatic speech recognition for human-robot interaction using an under-resourced language. Master’s thesis, Aalto University School of Electrical Engineering, Espoo.
- Mart Lubbers and Francisco Torreira. 2013-2015. pympi-ling: a python module for processing elans eaf and praats textgrid annotation files. <https://pypi.python.org/pypi/pympi-ling>. Version 1.69.
- André Mansikkaniemi, Peter Smit, Mikko Kurimo, et al. 2017. Automatic construction of the Finnish parliament speech corpus. In *INTERSPEECH 2017–18th Annual Conference of the International Speech Communication Association*.
- Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kald. In *Interspeech*, volume 2017, pages 498–502.
- Einar Meister, Lya Meister, and Rainer Metsvahi. 2012. New speech corpora at IoC. In *XXVII Fonetikaan päivät 2012 — Phonetics Symposium 2012: 17–18 February 2012, Tallinn, Estonia: Proceedings*, pages 30–33. TUT Press.
- Rachel M Olsen, Michael L Olsen, Joseph A Stanley, Margaret EL Renwick, and William Kretzschmar. 2017. Methods for transcription and forced alignment of a legacy speech corpus. In *Proceedings of Meetings on Acoustics 173EAA*, volume 30, page 060001. Acoustical Society of America.
- Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

- Mark A Pitt, Keith Johnson, Elizabeth Hume, Scott Kiesling, and William Raymond. 2005. The buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability. *Speech Communication*, 45(1):89–95.
- Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, CONF. IEEE Signal Processing Society.
- Tuomo Raitio, Antti Suni, Hannu Pulakka, Martti Vainio, and Paavo Alku. 2008. Hmm-based finnish text-to-speech system utilizing glottal inverse filtering. In *Ninth Annual Conference of the International Speech Communication Association*.
- Ingrid Rosenfelder, Josef Fruehwald, Keelan Evanini, and Jiahong Yuan. 2011. Fave (forced alignment and vowel extraction) program suite. [Http://fave.ling.upenn.edu](http://fave.ling.upenn.edu).
- Peter Smit, Sami Virpioja, and Mikko Kurimo. 2021. Advances in subword-based hmm-dnn speech recognition across languages. *Computer Speech & Language*, 66:101158.
- Jan Strunk, Florian Schiel, Frank Seifart, et al. 2014. Untrained forced alignment of transcriptions and audio for language documentation corpora using webmaus. In *LREC*, pages 3940–3947.
- Kevin Tang and Ryan Bennett. 2019. Unite and conquer: Bootstrapping forced alignment tools for closely-related minority languages (mayan). In *Proceedings of the 19th International Congress of Phonetic Sciences, Melbourne, Australia*, pages 1719–1723.
- Martti Vainio. 2001. Artificial neural network based prosody models for finnish text-to-speech synthesis.
- Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. 2006. Elan: a professional framework for multimodality research. In *5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1556–1559.
- Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. 2002. The htk book. *Cambridge university engineering department*, 3(175):12.
- Jiahong Yuan, Wei Lai, Chris Cieri, and Mark Liberman. 2018. Using forced alignment for phonetics research. *Chinese Language Resources and Processing: Text, Speech and Language Technology*. Springer.

Boosting Neural Machine Translation from Finnish to Northern Sámi with Rule-Based Backtranslation

Mikko Aulamo, Sami Virpioja, Yves Scherrer, Jörg Tiedemann

Department of Digital Humanities
University of Helsinki, Helsinki / Finland
{name.surname}@helsinki.fi

Abstract

We consider a low-resource translation task from Finnish into Northern Sámi. Collecting all available parallel data between the languages, we obtain around 30,000 sentence pairs. However, there exists a significantly larger monolingual Northern Sámi corpus, as well as a rule-based machine translation (RBMT) system between the languages. To make the best use of the monolingual data in a neural machine translation (NMT) system, we use the backtranslation approach to create synthetic parallel data from it using both NMT and RBMT systems. Evaluating the results on an in-domain test set and a small out-of-domain set, we find that the RBMT backtranslation outperforms NMT backtranslation clearly for the out-of-domain test set, but also slightly for the in-domain data, for which the NMT backtranslation model provided clearly better BLEU scores than the RBMT. In addition, combining both backtranslated data sets improves the RBMT approach only for the in-domain test set. This suggests that the RBMT system provides general-domain knowledge that cannot be found from the relative small parallel training data.

1 Introduction

Machine translation from and to minority languages is challenging because large parallel corpora are typically hard to obtain. Two strategies have proven most successful to eliminate this bottleneck: using rule-based machine translation (RBMT) systems that do not rely on large data, or training data-driven translation systems with automatically created synthetic data, e.g. backtranslation (Sennrich et al., 2016). In this paper, we com-

bine both strategies in the context of neural machine translation (NMT) from Finnish to Northern Sámi. In particular, we investigate the impact of RBMT in data augmentation in comparison to standard NMT-based backtranslation.

Northern Sámi is a Uralic minority language spoken in Norway, Sweden and Finland. Historically, most of the work on machine translation from and to Sámi languages is based on RBMT (Trosterud and Unhammer, 2012; Antonsen et al., 2017; Pirinen et al., 2017). Data-driven approaches such as NMT are generally more competitive, but require large amounts of training data in the form of parallel translated sentences. For minority languages, finding parallel data sets is usually more difficult than collecting monolingual data, which is also the case for Northern Sámi.

A common way of leveraging monolingual data for NMT is the above mentioned backtranslation strategy, a method where monolingual data of the target language is translated automatically to the source language to create additional parallel training data. In this work, we use two reverse translation models to produce the backtranslations: a neural model trained only on the available parallel data and a rule-based approach. The latter is a system developed for the translation from Northern Sámi to Finnish (Pirinen et al., 2017) within the Apertium framework (Forcada et al., 2011). We also combine both methods to further augment the data. Our experiments demonstrate the positive effects of both strategies and the possibility of obtaining complementary information from different backtranslation engines.

2 Related work

Using backtranslations from different sources as training data has been shown to be beneficial for improving machine translation quality. In addition to proposing training data augmentation methods that do not require reverse translation systems,

Burlot and Yvon (2018) compare the effects of using statistical machine translation (SMT) and NMT based backtranslations for English→French and English→German translations. They show that both types of backtranslations improve translation quality, NMT slightly more than SMT. Poncelas et al. (2019) also produce backtranslations with SMT and NMT. They show that the translation quality of a German→English NMT system is improved when including either type of backtranslations in the training data. The greatest improvement is observed when both types of backtranslations are used.

Augmenting training data with RBMT backtranslations has also proven to be useful for boosting translation quality. Dowling et al. (2019) use RBMT backtranslations to improve statistical machine translation performance for Scottish Gaelic→English translations. The authors show that backtranslations can be beneficial even in cases where the translation quality of the MT system used to produce the backtranslations is low. Soto et al. (2019) study the performance of NMT systems trained with augmented training data backtranslated using RBMT, SMT and NMT. They experiment with Basque→Spanish translations and show that the translation performance improves when using each type of augmented training data individually. Soto et al. (2020) also analyze the effects of using augmented training data backtranslated with the three different paradigms. They focus on two language pairs: a low-resource language pair, Basque→Spanish, and a high-resource language pair, German→English. In addition to showing similar results as Soto et al. (2019), they show further improvement in translation performance when all types of augmented training data are combined.

3 Data

The UiT freecorpus¹ contains a Finnish - Northern Sámi (fin-sme) parallel corpus with 110k sentence pairs and a distinct set of 868k monolingual Northern Sámi sentences. The UiT corpora are collected from multiple sources and cover various domains. Both the parallel and the monolingual corpora contain considerable amounts of duplicate lines. In this section, we describe our data cleaning and filtering efforts and the data split. For ad-

¹<https://giellatekno.uit.no/>

ditional evaluation, we collected a small test set consisting of translated YLE news articles².

Data filtering and cleaning is carried out with the OpusFilter toolbox (Aulamo et al., 2020). Our OpusFilter configuration files are available online³, which helps to replicate the data preprocessing steps. First, we remove duplicate lines from the parallel corpus. This process removes 67.7% of the sentence pairs, leaving us with 35,426 unique sentence pairs. The remaining data set is then cleaned with a set of filters from OpusFilter. Similar filtering setups have been confirmed to improve translation quality (Vázquez et al., 2019; Aulamo et al., 2020). In particular, we remove sentence pairs that satisfy one of the following conditions:

- One or both of the sentences are empty or longer than 100 words,
- The ratio of the sentence lengths in words is greater than 3,
- The sentence pair contains words longer than 40 characters,
- The sentence pair contains HTML elements,
- The sentences have dissimilar numerals based on the “Non-zero numerals score” (Vázquez et al., 2019),
- The sentences have dissimilar punctuation based on the “Terminal punctuation score” (Vázquez et al., 2019),
- The sentence pair contains characters outside of the Latin script,
- The sentences are not recognized to be their correct language by the `langid.py` language identifier (Lui and Baldwin, 2012).

After filtering, 29,106 clean sentence pairs remain in the parallel data set. From this clean set, 2000 pairs are randomly selected to form a validation set and another 2000 pairs to form a test set, leaving 25,106 pairs for training. Note that all subsets are disjoint due to the initial deduplication.

The additional test set consists of two news articles describing Sámi culture in Finland available in both Finnish and Northern Sámi on YLE News. It was extracted from the web and manually aligned to create a clean reference set. This test set

²<https://yle.fi/uutiset/osasto/sapmi/>

³<https://github.com/Helsinki-NLP/Sami-MT>

is, however, small (151 sentence pairs) and may not produce completely reliable evaluation scores, but it should still provide additional insights about the quality of the translation models and their ability to generalize to new domains.

The monolingual Northern Sámi data is processed in a similar way as the parallel data above. Duplicate removal discards 35.6% of the total of 867,677 sentences, leaving 559,074 sentences in the data set. For corpus cleaning, we use all filters of those cited above that are applicable to monolingual data, i.e. the sentence length filter, the word length filter, the HTML element filter, the Latin script filter, and the language identification filter. The resulting clean monolingual corpus contains 462,803 sentences.

4 Method

In this section, we compare a baseline fin-sme NMT model trained only with the available parallel data to NMT models trained with additional backtranslated data. The backtranslations are produced by translating the clean monolingual Northern Sámi data to Finnish either with a NMT system trained on the parallel data in the reverse direction (sme-fin), or with the sme-fin RBMT system. This yields three additional synthetic training sets that augment the original parallel training data: one with the NMT backtranslations, one with RBMT translations, and one with both types of backtranslations. Each of them is then used to train a separate NMT model that we can compare to the baseline model, which is trained on the original parallel data only. Note that we do not use any data sampling or weighting scheme to balance original and augmented training data.

All NMT models in our experiments are trained with MarianNMT (Junczys-Dowmunt et al., 2018) version 1.8.33. The backtranslation model is based on a RNN architecture with GRU cells (Cho et al., 2014) and attention. In our experiments, the RNN architecture slightly outperformed Transformers in the out-of-domain test set for this translation direction. All models using additional backtranslated training sets are trained with both RNNs and Transformers. All RNN models have the same architecture as the backtranslation model. For Transformers, we use the example hyperparameters from MarianNMT⁴ which replicate the setup

⁴<https://github.com/marian-nmt/marian-examples/tree/master/transformer>

	UiT	YLE
NMT	19.4	4.5
RBMT	12.3	10.0

Table 1: Reverse translation model (sme-fin) quality in BLEU points evaluated with the UiT test set and the YLE test set.

from Vaswani et al. (2017). For subword segmentation, we use the SentencePiece tokenizer (Kudo and Richardson, 2018) with vocabulary size 8000, which has been shown to produce the best results with the data set sizes that we are dealing with (Gowda and May, 2020; Grönroos et al., 2021). We train the models until the cross-entropy of the validation set does not improve for 10 consecutive validation steps.

For the RBMT backtranslations, we use Aperitium with the sme-fin model by Pirinen et al. (2017). This system implements a shallow transfer-based translation engine consisting of modules for morphological analysis, disambiguation and generation, modules for lexical translation based on context rules, and a module for syntactic transformation operations.

Table 1 shows the quality of the sme-fin translation models used for backtranslations in BLEU points (Papineni et al., 2002). The NMT model performs much better with UiT test data than with the YLE test data, which shows that the NMT system is strongly adapted to the UiT data, while the RBMT system has similar performance with both test sets.

4.1 Backtranslations

All the 462,803 sentences of the cleaned monolingual data are translated with the sme-fin NMT and RBMT models. As the quality of the source side of the backtranslations is not as important as the quality of the target side (Sennrich et al., 2016), we keep an unfiltered version of both backtranslation data sets. To see the effect of filtering the augmented data set, we apply OpusFilter with a reduced set of filters (recall that the monolingual Northern Sámi data has already been processed): sentence length filter, length ratio filter, word length filter, HTML element filter, non-zero numeral filter and terminal punctuation filter. After filtering and an additional deduplication step, the NMT-produced backtranslations amount to 415,313 sentence pairs and the RBMT-

	Training data	Transformer		RNN	
		UiT	YLE	UiT	YLE
Baseline	25,106	18.9	4.3	18.5	5.1
+ NMT-all-bt	470,085	32.9	9.2	23.0	8.4
+ RBMT-all-bt	487,862	37.0	14.4	26.4	11.0
+ NMT-all-bt + RBMT-all-bt	932,790	38.8	10.9	26.3	9.6
+ NMT-clean-bt	422,596	34.0	9.8	25.0	8.8
+ RBMT-clean-bt	378,567	36.3	15.5	25.6	10.9
+ NMT-clean-bt + RBMT-clean-bt	776,006	38.9	11.3	28.2	10.7
+ NMT-clean-bt + RBMT-all-bt	885,301	40.1	10.8	29.9	9.9

Table 2: Training data sizes (sentence pairs) and results (in BLEU points) for the fin-sme translation models with two different architectures (Transformer and RNN) using original parallel data (Baseline), augmented data sets with unfiltered and filtered backtranslations (all-bt and clean-bt, resp.) evaluated on the UiT test set and the YLE test set.

produced ones to 353,465 sentence pairs. After concatenation with the parallel data and removal of duplicates in this concatenated set, we are left with 422,596 and 378,567 sentence pairs respectively. Furthermore, another training set is created by merging both the NMT and RBMT backtranslations with the parallel data; this set contains 776,006 sentence pairs. The first column of Table 2 shows the training data sizes of the different configurations.

5 Results

The upper part of Table 2 shows the BLEU scores of the translation models trained with the original parallel data set (baseline) and the unfiltered augmented data sets. Similarly to the reverse model, the baseline fin-sme models are well adapted to the UiT test set and do not perform as well with the YLE test set. Adding the NMT backtranslations to the training data gives a significant improvement with respect to BLEU scores: using Transformers on the UiT set, the score raises by 14 points (74% relative), and on the YLE set, the score goes up by 4.9 points (114%). The RBMT backtranslations give an even larger boost on the UiT set than the NMT translations (18.1 points, 96%) and especially on the YLE data (10.1 points, 235%). Using RNNs, the scores are lower overall, but they do show similar improvements with the same training sets as Transformers.

The significant boost from RBMT backtranslations is quite remarkable considering that Apertium does not seem to perform very well on the reverse translation direction on UiT data. This result stresses once more that the effect of backtransla-

tion is to a larger extent due to improved target language coverage than to the quality of the translations. Instead, the additional, less domain-specific knowledge encoded in the RBMT model seems to lead to the additional push even in the UiT domain and it certainly carries over to the out-of-domain data represented by the YLE news data.

The simple combination of both types of backtranslations only provides a modest additional boost on the UiT test set. The out-of-domain performance drops substantially compared to using RBMT-based backtranslations alone. Adding NMT-based translations seem to hurt the model in this regard.

Next, we study the effect of filtering the backtranslations before training the augmented NMT models. Table 2 also shows the results of this approach. We can see that the models benefit from filtering the NMT backtranslations, especially on the UiT domain, whereas the RBMT-based augmentation model performance decreases on the UiT test set. The RBMT-based Transformer model gains an improvement on the YLE set, but the same score with the RNN model decreases slightly. The combination of both backtranslation augmentations leads to a boost in translation quality over the unfiltered backtranslation training set, which suggests that a careful data selection can be important when using data augmentation techniques. The performance on the YLE data is still lower than the RBMT-based data augmentation alone, which could indicate that the RBMT backtranslations are able to carry over out-of-domain information, but this result needs to be taken with a grain of salt as the test set is very small.

Finally, we also train a models that combine filtered NMT backtranslations with unfiltered RBMT backtranslations (last row in Table 2). These models reach the overall highest BLEU scores on the UiT test set, 40.1 with Transformer, but on the YLE test set the performance is lower than with other models, which is a bit surprising but may also depend on random variation and on the small size of the test set.

6 Conclusion

In this work, we confirm that the addition of backtranslations produced with multiple paradigms, including RBMT, improves the quality of NMT models. Additionally, the translation performance can be further improved by removing noisy sentence pairs from the NMT backtranslations. We show that these methods are beneficial in a real-world low-resource setting with the Finnish→Northern Sámi translation pair.

In the future, we plan to extend our work in various ways including more careful data selection and filtering, the use of subword regularization, domain labeling, improved sampling strategies and further data augmentation techniques such as pivot-based translations and transfer learning using multilingual NMT models. Furthermore, we would like to optimize hyper-parameters such as vocabulary size, network architectures and training parameters to maximize the translation performance in low-resource scenarios.

Acknowledgements

The research presented in this paper was supported by the European Language Grid project through its open call for pilot projects. The European Language Grid project has received funding from the European Union’s Horizon 2020 Research and Innovation programme under Grant Agreement N^o 825627(ELG). This work was also supported by the FoTran project, funded by the European Research Council (ERC) under the European Union’s Horizon 2020 Research and Innovation programme under Grant Agreement N^o 771113. The authors wish to acknowledge CSC – IT Center for Science, Finland, for computational resources.

References

Lene Antonsen, Ciprian Gerstenberger, Maja Kappfjell, Sandra Nystø Rahka, Marja-Liisa Olthuis,

Trond Trosterud, and Francis Tyers. 2017. Machine translation with North Saami as a pivot language. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 123–131.

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A configurable parallel corpus filtering toolbox. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 150–156, Online. Association for Computational Linguistics.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155, Brussels, Belgium. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Meghan Dowling, Teresa Lynn, and Andy Way. 2019. Leveraging backtranslation to improve machine translation for Gaelic languages. In *Proceedings of the Celtic Language Technology Workshop*, pages 58–62.

Mikel L Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O’Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144.

Thamme Gowda and Jonathan May. 2020. Finding the optimal vocabulary size for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3955–3964, Online. Association for Computational Linguistics.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2021. Transfer learning and subword sampling for asymmetric-resource one-to-many neural translation. *Machine Translation*, pages 1–36.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

- Taku Kudo and John Richardson. 2018. Sentence-Piece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Tommi Pirinen, Francis M. Tyers, Trond Trosterud, Ryan Johnson, Kevin Unhammer, and Tiina Puolakainen. 2017. North-Sámi to Finnish rule-based machine translation system. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 115–122, Gothenburg, Sweden. Association for Computational Linguistics.
- Alberto Poncelas, Maja Popović, Dimitar Shterionov, Gideon Maillette De Buy Wenniger, and Andy Way. 2019. Combining SMT and NMT back-translated data for efficient NMT. In *12th International Conference on Recent Advances in Natural Language Processing, RANLP 2019*, pages 922–931. Incoma Ltd., Shoumen, Bulgaria.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- Xabier Soto, Olatz Perez-De-Viñaspre, Maite Oronoz, and Gorka Labaka. 2019. Leveraging SNOMED CT terms and relations for machine translation of clinical texts from Basque to Spanish. In *Proceedings of the Second Workshop on Multilingualism at the Intersection of Knowledge Bases and Machine Translation*, pages 8–18, Dublin, Ireland. European Association for Machine Translation.
- Xabier Soto, Dimitar Shterionov, Alberto Poncelas, and Andy Way. 2020. Selecting backtranslated data from multiple sources for improved neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3898–3908. Association for Computational Linguistics.
- Trond Trosterud and Kevin Brubeck Unhammer. 2012. Evaluating North Sámi to Norwegian assimilation RBMT. In *Proceedings of the Third International Workshop on Free/Open-Source Rule-Based Machine Translation (FreeRBMT 2012)*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Raúl Vázquez, Umut Sulubacak, and Jörg Tiedemann. 2019. The University of Helsinki submission to the WMT19 parallel corpus filtering task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 294–300, Florence, Italy. Association for Computational Linguistics.

Building a Swedish Open-Domain Conversational Language Model

Tobias Norlund

Chalmers University of Technology
Recorded Future
tobiasno@chalmers.se

Agnes Stenbom

Royal Institute of Technology
Schibsted Media Group
astenbom@kth.se

Abstract

We present on-going work of evaluating the, to our knowledge, first large generative language model trained to converse in Swedish, using data from the online discussion forum Flashback. We conduct a human evaluation pilot study that indicates the model is often able to respond to conversations in both a human-like and informative manner, on a diverse set of topics. While data from online forums can be useful to build conversational systems, we reflect on the negative consequences that incautious application might have, and the need for taking active measures to safeguard against them.

1 Introduction

Dialog is an important means through which machines can exhibit intelligence toward humans, which is interesting from a general AI perspective. But dialog also constitutes a natural interface for humans to interact with technology, which opens up for a breadth of applications involving complex information acquisition, automation of tasks and smart support systems. A promising direction towards this goal is the development of open domain conversational systems using large neural networks.

Early approaches to neural conversational systems rely on various forms of Recurrent Neural Networks (RNN) trained autoregressively to model the textual sequences (Shang et al., 2015; Vinyals and Le, 2015; Sordani et al., 2015; Serban et al., 2016). More recently, as large pre-trained Transformer networks have come to dominate progress in NLP in general (Devlin et al., 2019; Radford, 2018; Radford et al., 2019; Brown et al., 2020; Raffel et al., 2020), approaches such as DialoGPT (Zhang et al., 2020), Meena (Adwardana et al., 2020) and Blender (Roller et al.,

2020) have proven the architecture’s applicability in open domain dialog systems as well.

However, as the research effort is predominantly put into making progress on English, the importance of making progress in other languages as well has been noted (Ruder, 2020; Wali et al., 2020). Each language is its own unique challenge for many reasons, but the difference in availability of resources is a major one, in particular for data-driven methods. We argue this is also important to keep the public debate on the risks and ethical aspects of large scale language models open to non-English speaking communities. Toward those ends, we present the first (to our knowledge) attempt to build a large scale open domain dialog system in Swedish based on data from Flashback, one of the largest social discussion forums in Sweden. We also present early indicative results on a human evaluation to assess its response generation capabilities across a wide range of topics.

2 Data and preprocessing

Flashback¹ is a Swedish online forum that launched in 1996 and has since grown to become one of the country’s most popular social medias (Internetstiftelsen, 2019). In the various sub forums, a breadth of topics are openly discussed including computers and programming, economics, politics, sports and science. To the general public however, the forum is also widely known for housing an anonymous safe haven for controversial subjects such as prostitution, drugs and conspiracy theories (Östman and Aschberg, 2015). Due to its consistent popularity over the last two decades, it arguably today makes up Sweden’s biggest single source of general conversational text.

On Flashback, posts are chronologically organized into threads. In a single thread, the discussion is centered around a specific topic typically

¹<http://www.flashback.org>

Number of layers	48
Dimensionality	1600
Feed-forward dim	5400
Number of heads	16
Number of parameters	1.4B
Max context length	400
Batch size	512
Optimizer	Adam
Vocabulary size	52,000

Table 1: Model hyperparameters

described by a thread title. Acknowledging the potential for embedding undesired biases, we have initially chosen to use a complete and unfiltered dump of the forum for this study.

The data was tokenized into strings of BPE tokens (Sennrich et al., 2016) using a customly trained vocabulary. Due to Flashback’s organization of posts into a single linear feed (unlike the tree structure on e.g. Reddit), it is common that users quote the previous post they respond to, to avoid confusion. As a quote holds important contextual information to a post, we chose to explicitly include this in the way we formatted the threads. More details of how the data was formatted into strings can be found in Appendix A.

3 Model

Following previous works on open-domain dialogue systems (Zhang et al., 2020; Adiwardana et al., 2020), we trained an auto-regressive language model using a slightly modified Transformer (Vaswani et al.) decoder as proposed by Radford et. al. (2019). That is, for an input sequence of tokens x_1, \dots, x_n , the language model is trained to maximize the likelihood of the joint probability:

$$p(x_1, \dots, x_n) = p(x_1) \prod_{i=2}^n p(x_i | x_{i-1}, \dots, x_1) \quad (1)$$

We denote our model *Flashback-GPT*, where GPT is an acronym for Generative Pre-trained Transformer as first coined by Radford et. al. (2019). The hyper-parameters chosen are similar to those of the largest variant of GPT-2 (Radford et al., 2019), and are detailed in Table 1.

The model was trained on 16 Nvidia Tesla V100 SXM2 GPUs for 7 days, equivalent to 86,250 gradient updates. The learning rate was increased

linearly for the first 5,000 steps up until $5e-5$, after which it was kept constant. We used the `deepspeed` (Rasley et al., 2020) library to optimize memory efficiency across the devices during training.

4 Evaluation

Evaluating natural language generation systems is known to be hard. Even though it is common to conduct automatic evaluations due to their low cost, a human evaluation often serves as an additional validation of the results. However, designing a human evaluation to measure a specific quantity is also not trivial since there is always room for interpretation among the human annotators.

Therefore, we present a pilot study where the main aim is merely to get early indications rather than definite results, and to guide the design of bigger future studies. We design our pilot to measure our quantity of interest: To which extent is the model capable of participating in social discussion forums across a diverse set of topics?

To that end, we seek to measure two quantities: *humanlikeness* and *informativeness*. As language models can often be inconsistent and show lack of commonsense knowledge, humanlikeness is supposed to answer if there is anything in a response that seems off, suggesting it has not been written by a human. However, a response can be human-like but still uninformative. The notion of ”informativeness” is particularly interesting in our setting as forums can be relatively knowledge centric, and uninformative responses such as *I don’t know* add little to the discussion.

4.1 Study design

The study was designed as follows. We select a set of N Flashback threads, held out from training, to be used in the study. For each thread, we only take the first two or three posts to limit the discussion context. We then, for each thread, swap the last post for an alternative generated by the model. Along with the originals, we now have $2N$ threads that we present (in shuffled order) to human annotators. For each thread, we ask two binary questions to measure humanlikeness and informativeness respectively:

1. Is there any indication that the last message was not written by a human?
2. Do you think that the last message adds information to the discussion?

This draws close resemblance to previous evaluations performed on English systems (Zhang et al., 2020; Adiwardana et al., 2020). In Zhang et al. (2020), humans are asked to rank two alternative responses according to *informativeness*, *human-likeness* and *relevance*. In Adiwardana et al. (2020), humans are instead asked the binary questions whether a response "makes sense" and also whether it is "specific", and the average of the two (Sensibleness and Specificity Average - SSA) is found to correlate with humanlikeness. For simplicity, we chose to directly ask for humanlikeness instead of the SSA proxy questions. The complete annotator guideline (Swedish) is included in Appendix B for reference.

For the pilot study, we collected a sample of $N = 120$ Flashback threads, stratified across 12 of the top level forums. We then formed two groups of human annotators with three persons in each group. Each group was presented 60 threads with generated responses, and 60 original, with no overlap. The threads included were randomly chosen, except for a few criteria that we employed to prevent the annotators from exploiting obvious surface patterns when answering question 1.

- As has been noted previously (Roller et al., 2020), beam search decoding strategies have a tendency to generate shorter responses over longer. We decided to only include threads where the last (human written) response is at most 200 characters.
- Since the model supports a maximum sequence length of 400 tokens, we exclude threads where the context is longer than 350 tokens, to leave some room for the generated response.
- Since the model often fails to generate correct quotes of previous responses, we remove any quotes from the last (human written) response, and force the model not to generate quotes as well.

We include a subset of the threads (both with generated and ground truth responses) in Appendix C.

4.2 Decoding

The decoding strategy used to generate responses from neural language models is an important part of the system as a whole (Roller et al., 2020).

	Flashback-GPT	Human
Humanlike	68% (48%)	95% (79%)
Informative	48% (52%)	83% (74%)
Humanlike + informative	46%	83%

Table 2: Pilot study results. *Humanlike* is the percentage where the majority response to the first question is *no*. *Informative* is the percentage where the majority response to the second question is *yes*. Numbers in parentheses are percentages of the 120 threads where all three annotators agreed

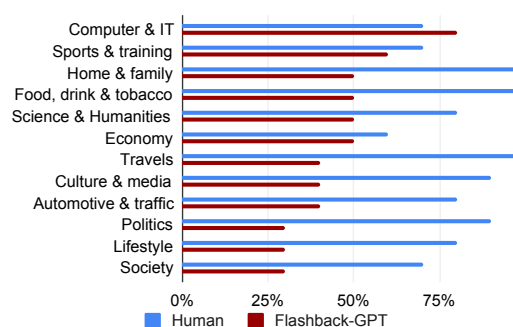


Figure 1: Ratio of responses that were deemed both humanlike and informative for each of the evaluated forums

While the commonly employed beam search algorithm is optimizing the joint likelihood for the whole generated sequence, its outputs are known to be generic, unspecific and repetitive (Holtzman et al., 2020; See et al., 2019). We chose to use a beam sampling strategy, where we at each step, for each beam, sample from the (re-normalized) top 50 predicted vocabulary items. This struck a good balance between generating short uninformative responses vs longer incoherent ramblings. We used a beam size of 6. The model has a tendency to generate responses such as "duplicate thread, locking //mod", which are commonly found on Flashback but are not very interesting for this study. We try to circumvent this by banning the generation of certain distinguishing words, such as "mod". Finally, to avoid repetitions we also prevent the model from generating repetitions of any 3-grams occurring in the context, or in the generated sequence thus far.

5 Results and Discussion

Results from the study are shown in Table 2. We judge a thread’s humanlikeness and informativeness based on the majority response from the three annotators. We also report the percentage of threads where all annotators agreed in their responses.

Unsurprisingly, ground truth human responses display a high ratio of humanlikeness, consolidated by a relatively high degree of annotator agreement. Our model’s responses also show signs of humanlikeness, as suggested by the fact that 68% of its generated responses were deemed plausible to be human-written. We note however that the annotator agreement is significantly lower compared to ground truth responses, suggesting we could further clarify the humanlikeness question we ask the human annotators.

The model shows less strength on our measure of informativeness, with only 48% of the model’s generated responses were deemed informative to the discussion. If we compare the amount of threads where the responses were both deemed humanlike and informative, the model’s ratio drops to 46% compared to 83% for the ground truth responses. While our sample size is too small to draw any statistically significant conclusions, Figure 1 shows the distribution of humanlike + informative responses over their top-level forums. Interestingly, the top-3 most popular forums (Society, Politics and Culture & media), which together comprise 41% of the training data, all perform below average.

Qualitative feedback from the annotators highlight how the model tends to respond with short and straight answers, less prone to vent thoughts and opinions compared to human responses. Common failure modes include completely misunderstanding the question being asked, or change of topic to a related but irrelevant one.

Reflecting on the design of the study, we found very few responses were deemed informative but not humanlike (2 of the generated, 0 ground truth). If the main purpose of a future study is to measure both humanlikeness and informativeness, the question of informativeness might be sufficient.

6 Broader implications

Conversational models such as that presented in this paper can be understood as part of a broader transformation of communication. As argued by

Guzman and Lewis (2020), we are now moving away from the traditional view of communication as anchored in human such. How we apply and evaluate conversational models going forward may come to alter the way we relate to each other as communicators, and ultimately, humans. There is need for informed discussion around what constitutes *desirable* use. While highlighting the risks of these emerging technologies could be considered detrimental, we believe it to be an important means towards enabling the inclusion of diverse perspectives in this discussion.

A prominent issue related to NLP is found in the notion of bias. Explicit and implicit biases concerning gender, race or disability can be embedded in e.g. text corpora (Caliskan et al., 2017), word embeddings (Bolukbasi et al., 2016) and generative models (Sheng et al., 2019). Employing biased conversational models risks scaling systematic discrimination of various groups in society.

When developing conversational technologies, we must acknowledge that they can be used for malicious purposes. As generative language technology improves and grows in Swedish, so will its ability to manipulate and deceive at scale. As noted by the Swedish Defence Research Agency (FOI), recent developments within generative language technology present risks of increased computer-generated false news and comments – predominately on social media – possibly posing a national security threat (Lundén et al., 2021).

Potential harm must also be considered on the individual level. In 2020, a GPT-3-powered (Brown et al., 2020) bot engaged in Reddit-forums with 30 million users about sensitive topics such as suicide and conspiracy theories (Heaven, 2020). With the indicative model performance demonstrated in this article, such human-machine communication could soon transpire in Swedish.

7 Conclusions and Future work

We demonstrate that Flashback can provide a base on which to build general conversational systems in Swedish. While our early results suggest the model is often capable to converse across a diverse set of topics, more work remains to examine its utility on various conversational tasks. We also believe developing methods for grounding the responses in additional data is an interesting direction to further the performance on in-

formativeness in particular. However, we also believe particular care should be taken as the underlying data is known to contain toxic content. This points to the importance of putting our model through further scrutiny in following work, to better understand its biases, how they are manifested in downstream tasks, and how they can be mitigated. Towards those ends, we intend to make the model available for such purposes, and more information is available at <https://github.com/TobiasNorlund/flashback-gpt>

Acknowledgments

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at Chalmers Centre for Computational Science and Engineering (C3SE) partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

References

- Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot.
- Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, page 4356–4364, Red Hook, NY, USA. Curran Associates Inc.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.
- Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. 356(6334):183–186.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics.
- Andrea L Guzman and Seth C Lewis. 2020. Artificial intelligence and communication: A human-machine communication research agenda. *New Media & Society*, 22(1):70–86.
- Will Douglas Heaven. 2020. A gpt-3 bot posted comments on reddit for a week and no one noticed.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration.
- Internetstiftelsen. 2019. Svenskarna och internet 2019.
- Jenny Lundén, Anders Melander, Elin Hellquist, Björn Ottosson, Liselotte Steen, and Anders Strindberg. 2021. Strategisk utblick 9 framtida hot.
- Alec Radford. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. New York, NY, USA. Association for Computing Machinery.
- Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, and Jason Weston. 2020. Recipes for building an open-domain chatbot.
- Sebastian Ruder. 2020. Why You Should Do NLP Beyond English. <http://ruder.io/nlp-beyond-english>.
- Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. In *Proceedings of the 2019 Conference of the North*

- American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany. Association for Computational Linguistics.
- Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 3776–3783. AAAI Press.
- Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China. Association for Computational Linguistics.
- Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The woman worked as a babysitter: On biases in language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics.
- Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008. Curran Associates, Inc.
- Oriol Vinyals and Quoc Le. 2015. A neural conversational model.
- Esma Wali, Yan Chen, Christopher Mahoney, Thomas Middleton, Marzieh Babaeianjelodar, Mariama Njie, and Jeanna Neefe Matthews. 2020. Is machine learning speaking my language? a critical look at the nlp-pipeline across 8 human languages.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. In *ACL, system demonstration*.
- Karin Östman and Richard Aschberg. 2015. Flashback – ett laglöst land.

Appendix A Flashback data details

The data needs to be converted into a textual string format for it to be compatible with a standard language model. To this end, each thread was formatted into textual *records*. Listing 1 provides an example of a formatted data record used to train the model. A record can be at most 400 tokens, and as such, threads are often broken up into multiple records. This means the model will in general not have the full thread context when predicting the next message.

```
1 Dator och IT > Hårdvara: PC
2 Luft eller vattenkylning till cpu
3
4 [user1]:
5 Jag har lite beslutsångest till vilken kylning jag ska satsa på till min AMD Phenom
  II X4 965 AM3.
6 Denna fläkten http://www.komplett.se/k/ki.aspx?sku=456730 eller är det smartare att
  satsa på vattenkylning?
7
8 [user2]:
9 Citat: [user1]
10 Jag har lite beslutsångest till vilken kylning jag ska satsa på till min AMD
  Phenom II X4 965 AM3.
11 Denna fläkten http://www.komplett.se/k/ki.aspx?sku=456730 eller är det
  smartare att satsa på vattenkylning?
12 Det där var väl ett jävla åbäk iaf, är du säker på att det inte finns bättre för typ
  halva priset? Typ Noctua eller liknande?
13
14 [user3]:
15 En vettig fråga är: Vad skall du göra med datorn? Extrem överklockning? Få en tyst
  dator?
```

Listing 1: Example of a formatted training record. The usernames are anonymized.

Table 3 details the amount of data from each of the top level forums that was used for training. The dump was collected in September 2020 and in total the data comprised 23.5 GB of raw formatted text.

Top-level forum (swedish)	Top-level forum (english)	Num threads	Num posts
Samhälle	Society	230,931	8,681,841
Politik	Politics	123,031	7,578,865
Kultur & Media	Culture & Media	165,929	6,495,860
Vetenskap & humaniora	Science & Humanities	225,139	5,130,519
Dator och IT	Computer & IT	334,931	4,833,468
Sport & träning	Sports & training	81,922	4,475,793
Hem, bostad & familj	Home & family	158,819	4,055,688
Droger	Drugs	137,870	3,551,768
Övrigt	Others	75,735	2,164,237
Livsstil	Lifestyle	81,750	2,060,600
Sex	Sex	49,512	1,335,657
Ekonomi	Economy	68,078	1,327,001
Mat, dryck & tobak	Food, drink & tobacco	51,133	1,286,707
Fordon & trafik	Automotive & traffic	68,078	1,070,619
Om Flashback	About Flashback	73,910	486,536
Resor	Travels	29,514	478,150
-	Forum unknown	181	71,933
Total		1,956,463	55,085,242

Table 3: Flashback training data statistics

Appendix B Annotation guideline for human evaluation

Annoteringsbeskrivning: Flashback

Den annotering som du skall genomföra är en del av ett forskningsprojekt för att studera en ny typ av chatbot. Chatboten är framtagen för att efterlikna människor i diskussionsforum.

Du kommer gå igenom ett kalkylark med diskussionsråd från internetforumet Flashback. Varje diskussionsråd innehåller 2 eller 3 meddelanden. För varje tråd förväntas du svara på två frågor som båda rör **det sista meddelandet i konversationen** (markerat med **grönt** nedan).

Ett exempel på en sådan tråd är:

Kultur & Media > Film och filmproduktion > Film: listor och rekommendationer
Någon tecknad film som är bättre dubbad på svenska?

KP19c88:

Ja som rubriken säger.

Fanns det någon tecknad film som du föredrar på svenska?

Eller något annat språk kanske?

Jag föredrar nog de flesta tecknade filmerna i sitt originalspråk men jag har nog märkt att den ende som står ut är nog Lejonkungen.

Tycker att osskådespelarna är bättre och mer rytmiserade än på engelska. Vad tycker du?

AnuroBandini:

Jag tror inte riktigt att jag kan svara helt objektivt på det, då mycket av glädjen i att se tecknade idag beror på minnen från dessa filmer som man hade när man var liten. Därför så skulle jag ha svårt att tänka mig att se typ Ducktales på engelska.

KP19c88:

Visst mysiska jag nog vara kvar från hur man såg det då. Men generellt sett tycker jag att det mesta är bättre på sitt originalspråk

Varje diskussionsråd börjar med det *förum* på Flashback som tråden är skriven i (markerat i **orange** ovan). Därefter följer trådens *rubrik* (markerat med **gul**). Sedan kommer ett antal *meddelanden*, där varje meddelande börjar med ett användarnamn+kolon (markerat med **blå**) och därefter ett antal textrader:

I kalkylarket finns två svars-kolumner. Vi vill att du för varje diskussionsråd svarar på följande frågor:

1. **Finns det något som tyder på att det sista meddelandet lite är skrivet av en människa?**
 - Exempel kan vara att den säger något felaktigt, är motsägelsefull eller generellt säger något som man inte förväntar sig av en Flashback-användare.

2. Tycker du att svaret tillför information till diskussionen?

- Om ditt svar är ja, skriv då "1" i svars-kolumnen. Annars skriver du "0"
- Syftet med denna fråga är att ta reda på *hur ofta chatboten skriver något som inte går att skilja från en människa?*
- Om du är osäker på grund av en faktauppgift i meddelandet som du ej vet är sann eller lögn i sammanhanget behöver du inte kontrollera denna genom att exempelvis googla, utan svara i sådana fall "0".
- Om det sista meddelandet enligt din mening inte tillför särskilt mycket till diskussionen, svara med "0" annars "1".
- Exempel på detta kan vara om meddelandet är orelaterat till ämnet t.ex. att en moderator skriver att hen läser tråden eller att det skrivs att det redan finns en tråd om ämnet. I sådana fall svarar du "0".
- Ett annat exempel kan vara om tråden handlar om hur man löser en matematisk ekvation. Då tillför ett svar såsom "Lös ekvationen" inte särskilt mycket till diskussionen, i vilket fall du också svarar "0".

Kalkylarket innehåller diskussionsråd där det sista meddelandet antingen är automatiskt genererat eller ett fiktiskt Flashback-meddelande.

Efter att du svarat på alla diskussionsråd i kalkylarket, vänligen sammanfatta i några få meningar vad som du tycker är utmärkande för chatboten (som fått dig att svara "1" på fråga 1).

Appendix C Examples from study

In the following examples, the last response is generated by the model. Usernames are anonymized.

```
1 Fordon & trafik > Motorcyklar och mopeder
2 Off road MC
3
4 [user1]:
5 När jag blir äldre vill jag köra Off road MC, typ Yamaha WR250X verkar nice.
6 Annars finns det yamaha XT125X. men med mindre klenare motor.
7 Det är ju bäst att skaffa mc kort med obegränsad motoreffekt, så jag för köra alla
  typer.
8 Jag är 175 cm just nu.
9 1. Hade jag kunnat ha en Yamaha WR250X, så jag inte är för kort?
10 någon som vet?
11
12 [user2]:
13 när du blir äldre?
14 e du över 18..?
15 men nej, tror inte du e för kort att köra off road mc
16
17 [user3]:
18 Vad ska du ha den till?
```

```
1 Vetenskap & humaniora > Fysik, matematik och teknologi > Matematiska och
  naturvetenskapliga uppgifter
2 ekvationer som omformas med formler-ma d
3
4 [user1]:
5  $5\sin 4x = 3\sin 2x$ 
6 lös ekvationen och svara med en decimal?
7 Jag vet att jag ska flytta över HL i VL. Men sedan vet jag ej vad jag ska göra.
8
9 [user2]:
10  $\sin(2a) = 2\sin(a)\cos(a)$  giver ju i princip svaret.
11
12 [user3]:
13 Du skall multiplicera båda leden med  $\cos(a)$ .
```

```
1 Resor > Övriga resediskussioner
2 Beställa saker som ligger i planet när jag sätter mig!
3
4 [user1]:
5 Heellu, har en fråga här.. När man beställer saker ifrån tax-free saken så det
  ligger i sätet när man kommer in i planet, måste man vara 20 och över för vodka
  o sånt då?
6 Är inte mer än 19 när jag ska åka, vore gûtt att få med sig en flaska ner dit man nu
  ska =)
7
8 [user2]:
9 Eftersom du är 19 så misstänker jag att du ska ner till nåt varmt partyställe runt
  medelhavet, har jag rätt? I så fall är det billigare att köpa den där flaskan på
  plats och de bryr sig inte om din ålder.
10
11 [user3]:
12 Du behöver inte vara 20 för att köpa sprit i tax-freen.
```

Below are the same examples, but translated to English

```
1 Automotive & traffic > Motorcycles and mopeds
2 Off road MC
3
4 [user1]:
5 When I get older I want to drive Off road MC, like Yamaha WR250X seems nice.
6 Otherwise there is yahama XT125X. but with a weaker engine.
7 It is best to get the mc license with unlimited power, so I can drive all types.
8 I'm 175cm right now.
9 1. Can I have a Yamaha WR250X, or am I too short?
10 anyone who knows?
11
12 [user2]:
13 when you get older?
14 are you above 18..?
15 but no, don't think you're too short to drive off road mc
16
17 [user3]:
18 What are you gonna use it for?
```

```
1 Science & Humanities > physics, mathematics and technology > Mathematical and
  natural science exercises
2 reshaping equations with forumlas-ma d
3
4 [user1]:
5  $5\sin 4x = 3\sin 2x$ 
6 solve the equation and answer with one decimal?
7 I know I should move right-side over to left-side. But then I don't know what to do.
8
9 [user2]:
10  $\sin(2a) = 2\sin(a)\cos(a)$  basically gives you the answer
11
12 [user3]:
13 You should multiply both sides with  $\cos(a)$ .
```

```
1 Travels > Other travel discussions
2 Order things to my plane seat
3
4 [user1]:
5 Heellu, got a question here.. When you order stuff from the tax-free thing they lie
  on your seat when you board the plane, do you have to be 20 or above for vodka
  and such then?
6 Won't be more than 19 when I'm going, would be sweet to bring a bottle down to the
  destination =)
7
8 [user2]:
9 Since you are 19 I'm suspecting you're going down to some warm party place around
  the Mediterranean, am I right? In such case it is cheaper to buy that bottle in-
  place and they won't care about your age.
10
11 [user3]:
12 You don't need to be 20 to buy spirits in the tax-free.
```

It’s Basically the Same Language Anyway: the Case for a Nordic Language Model

Magnus Sahlgren*
RISE
Sweden

Fredrik Olsson
RISE
Sweden

Fredrik Carlsson
RISE
Sweden

Love Börjeson
KB
Sweden

Abstract

When is it beneficial for a research community to organize a broader collaborative effort on a topic, and when should we instead promote individual efforts? In this **opinion piece**, we argue that we are at a stage in the development of large-scale language models where a collaborative effort is desirable, despite the fact that the preconditions for making individual contributions have never been better. We consider a number of arguments for collaboratively developing a large-scale Nordic language model, include environmental considerations, cost, data availability, language typology, cultural similarity, and transparency. Our primary goal is to **raise awareness** and **foster a discussion** about our potential impact and responsibility as NLP community.

1 Introduction

Deep Transformer language models have become the weapon of choice in modern NLP (and in AI more generally). There is a rich, and evergrowing, flora of models available, including BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), Electra (Clark et al., 2020), T5 (Raffel et al., 2020), and GPT (2 and 3) (Radford et al., 2019; Brown et al., 2020). These models present slight variations of architectural choices, training objectives, parameter settings, and size and composition of the training data. Despite some internal variation in performance, Transformer language models in general hold state of the art results in basically all NLP benchmarks and evaluation frameworks at the moment (Wang et al., 2018, 2019; Nie et al., 2020).

The downside to this recent development is the computational cost of training deep Transformer

models. Starting with BERT-Base with its (now viewed as modest, but at the time of publication seen as substantial) 110 million parameters, and BERT-Large with its 340 million parameters, there has been a virtual explosion in the number of parameters, culminating in the recent GPT-3 with its 175 billion parameters (Brown et al., 2020), GShard with 600 billion parameters (Lepikhin et al., 2021), and the most recent Switch Transformer with a whopping 1,3 *trillion* parameters (Fedus et al., 2021). This development translates into an acute need for access to powerful processing platforms, huge amounts of training data, and an ability to harbor extremely long training times. Taken together, this is a perfect recipe for extreme energy consumption and cost, which risks leading to reduced inclusivity in research on large-scale language models.

There is a budding debate on the environmental and cultural impact of training and using large-scale language models. Two recent examples are Strubell et al. (2019) and Bender et al. (2021); the former analyze the energy consumption and cost of training deep Transformer language models, and the latter voice concerns regarding both the environmental and cultural impact of training and using large-scale language models. We hope to contribute to this discussion by providing a Nordic perspective on the need for large-scale language models. We will assume the position that a collaborative effort towards training a large-scale Nordic language model is something worth striving for. We consider a number of arguments for this position, include environmental considerations, cost, data availability, language typology, cultural similarity, and transparency.

2 Argument 1: The Environment

Strubell et al. (2019) estimate that the CO₂ emission from training a single BERT-Base amounts to roughly 652 kg (1,438 lbs), which is comparable

Corresponding author: magnus.sahlgren@ri.se

to a flight between San Fransisco and New York, or the average emissions resulting from electricity and heating for one person for one year in Stockholm.¹ This is something of a best-case scenario; the authors also calculate that training a BERT-Large with neural architecture search emits something like 284 *tonnes* of CO₂, which is roughly equivalent to the emissions of 56 average persons, throughout a year. An interesting question thus becomes: how much CO₂ emission has been produced as a result of the current development in NLP?

It is of course impossible to get an accurate count on this, but one way to approximate an answer might be to consider how many models have been trained in the world so far. We obviously cannot know this either, but we might be able to get an idea by looking at the number of models published in open source libraries. Luckily, much of the recent development is centered around one such library: the Transformers library of the company Hugging Face.² The Transformers library contains (at the time of submission) more than 6,800 models covering a total of 250 languages. A survey carried out by Benaich and Hogarth in the fall of 2020 claims that more than 1,000 companies are using the Transformers library in production, and that it has been installed more than 5 million times.³

6,800 models times a low estimate of 652 kg of CO₂ sums to 4,434 tonnes of CO₂ emissions. This is of course an extremely unreliable estimate. Many of the models uploaded to the Transformers library are merely finetuned and not trained from scratch (we have not been able to quantify this proportion). On the other hand, many of the uploaded models are significantly larger than BERT-Base, and one can assume that only a fraction of models that are built are actually uploaded to the Transformers model repository. By comparison, the average Swedish citizen emits around 8 tonnes of CO₂ per year,⁴ while RISE (the Research Institutes of Sweden) with approximately 2,800 employees emitted a total of 1,287 tonnes CO₂ during 2019 according to the 2019 annual report.

Counting only the Nordic models uploaded to Hugging Face, there are (at the time of submis-

¹www.regionfakta.com

²<https://github.com/huggingface/transformers>

³<https://www.stateof.ai/> (slide 127)

⁴www.naturvardsverket.se

Language	Number of models
Swedish	215
Danish	43
Norwegian	33
Icelandic	28
Norwegian Bokmål	12
Norwegian Nynorsk	12
Faroese	11

Table 1: Number of language models available for the Nordic languages via Hugging Face’s Transformers library (at the time of submission).

sion) a total of 354 models for the Nordic languages (see Table 1). Based on the assumptions in Strubell et al. (2019), this amounts to more than 230 tonnes of CO₂. By comparison, Anthony et al. (2020) estimates (using slightly different assumptions than Strubell et al. (2019)) that training GPT-3 resulted in at least 85 tonnes of CO₂ emission. Although these estimates are not directly comparable, they indicate that a focused effort to produce a large-scale Nordic language model may lead to a smaller carbon footprint than the current development where we see a steady increase in the number of monolingual models.

3 Argument 2: Cost

It is anything but cheap to train large-scale language models. The cost for performing a single training pass for the largest T5 model is estimated to be \$1, 3 million (Sharir et al., 2020), while training GPT-3 is estimated at around \$4, 6 million.⁵ To put these numbers into perspective, the average project funding in the EU Horizon 2020 program is estimated to be around \$2, 1 million,⁶ while the average national research project is typically not more than around \$150 *thousand*.⁷ This means that, unless you happen to be in the possession of a sizeable computing infrastructure, training models on this scale will be out of the question for most researchers.

However, even with access to suitable GPUs, it is not obvious that it will be possible to train a model on the required scale. Li (2020) estimates

⁵lambdalabs.com/blog/demystifying-gpt-3/

⁶accelopment.com/blog/lessons-learned-from-horizon-2020-for-its-final-2-years/

⁷vr.se/soka-finansiering/beslut/2020-09-08-humaniora-och-samhallsvetenskap.html

that performing a single training run with the full GPT-3 using an NVIDIA Tesla V100 GPU at its theoretical max speed would require 355 years. Assuming access to an NVIDIA DGX-1, which features 8 V100 GPUs, we would still need 44 years to build a replica of GPT-3. The cost of buying a DGX-1 machine is around \$129 thousand – i.e. roughly the size of an average national research project.

The sizeable cost (monetary as well as temporal) required to build a large-scale language model effectively excludes a large proportion of the NLP community from training models. This may not be entirely negative, considering the environmental concerns raised in the previous section, but it would be desirable if the production of large-scale language models was more inclusive and collaborative, with transparency and the possibility to influence the procedure even by smaller research groups. A communal effort would not only enable more researchers to have an influence on the model design, but it may also lead to broader usage of the resulting model, thereby reducing the need to constantly build new small (and probably not very useful) models.

4 Argument 3: Data Size and Transfer

It is a known fact that bigger training data leads to improved performance when using statistical learning methods in NLP (Banko and Brill, 2001; Sahlgren and Lenci, 2016). This has been eminently well demonstrated in the context of language models by the recent improvements using models that have been trained on very large data samples (Raffel et al., 2020; Brown et al., 2020). It is a fascinating question whether there *at all* exists sufficiently large text data to build native models for all Nordic languages.

Considering the biggest Nordic language Swedish as an example, Sweden has legal deposit laws installed in 1661 for everything printed. During the the twentieth century it was gradually extended to include sound, moving images and computer games and electronic material. The law for legal deposit of electronic material was added in 2012. As a result, the National Library of Sweden (KB), has vast and ever growing collections, closing in on 26 Petabyte of data.

Though only a fraction of the collections are digitized, the digital collections are nonetheless substantial. KB, through its data lab

(KBLab), works continuously to assemble corpora of Swedish texts and to make them available for modeling. The latest corpus of cleaned, edited, raw Swedish text is just over 104 GB of size (corresponding to approximately 1,4 billion sentences and 18,2 billion words). The sources for this corpus are: Swedish Wikipedia 2 GB; Governmental texts 5 GB; Electronic publications 0,4 GB; Social media 5GB; Monographs 2GB, and; Newspapers 90 GB. The corpus currently under construction increases primarily the share of born digital text from legal electronic deposits and is expected to be around 1 TB of cleaned, edited, raw Swedish text (thus approximately 14 billion sentences and 182 billion words). The upper limit (in terms of size) for subsequent corpora is expected to be between 2–5 TB, depending on the possibilities to transcribe spoken Swedish present in the KB collections.

The situations in the other Nordic countries are similar, relative to the size of the population in the respective countries. There are consequently extensive Danish and Norwegian collections available, whereas the text/data resources in Iceland and Faroe Islands are expected to be substantially smaller. Combining *all* Nordic text resources would likely lead to a fairly substantial data source, likely on the order of Terabytes.

The data conditions for the larger Nordic languages look promising even when considered individually, but it is not obvious that there even exists enough data to train native large-scale models in the smaller Nordic languages. Fortunately, it has been demonstrated that multilingual models improve the performance for languages with less available training data, due to transfer effects (Conneau et al., 2020). In particular, the transfer effects seems to be specifically beneficial for typologically similar languages (Karthikeyan et al., 2020; Lauscher et al., 2020). It is thus likely that in particular Icelandic and Faroese would benefit from a joint Nordic language model.

5 Argument 4: Typology

The Nordic languages belong to one of three Germanic language groups, also referred to as North Germanic languages (in addition to West and now extinct East Germanic). The North Germanic language group is further divided into two branches: East Scandinavian languages, which includes Swedish and Danish, and West Scandina-

vian languages, which contains Norwegian, Icelandic and Faroese. This genealogical categorization is sometimes contrasted with a distinction based on mutual intelligibility, which separates Continental Scandinavian (Swedish, Norwegian and Danish) from Insular Scandinavian (Icelandic and Faroese).

The Nordic languages are so similar from a typological perspective that the language boundaries have been, if not in dispute, at least subject to some discussion (Stampe Sletten et al., 2005). The difference between dialects *within* the Nordic languages is in some cases probably larger than the difference *between* the languages. A telling example is the difference between Norwegian Bokmål, which is very similar to Danish and as such is categorized as an East Scandinavian language, and Nynorsk, which is categorized as a West Scandinavian language. Another example is the difference between Jamtlandish (or Jamska, a dialect spoken in the Swedish region Jämtland, which is categorized as a West Scandinavian language) and standard Swedish (which is East Scandinavian).

From a typological perspective, it thus makes sense to entertain the idea of a joint North Germanic language model, in particular when considering the potential for transfer effects to the smaller Nordic languages. Of course, one can always ask whether we should not aim for a combined Germanic model instead? There will probably be something like an order of magnitude more data available if we consider *all* Germanic languages rather than just the Nordic ones. However, one can expect diminishing returns by adding more data at some point, and it is an interesting (and, as far as we are aware, open) question what is the trade-off between language similarity and data size?

6 Argument 5: Culture

Bender et al. (2021) raise concerns about the considerable anglocentrism of current language models. We agree that this is potentially problematic; most current models are trained on data harvested from the Internet, which we know is produced by certain demographics, and as such is not representative of the general population.⁸ A consequence of this is that current language models only encode the perspectives of certain groups of people, and

⁸<https://www.pewresearch.org/internet/fact-sheet/social-media/>

these people tend to *not* belong to marginalized groups. It is well-known that language models encode biases and prejudice that may be problematic (Bordia and Bowman, 2019; May et al., 2019).

Anglocentrism is not necessarily a disqualifying factor for the Nordic countries, some of which (such as Sweden) is sometimes considered to be among the most Americanized countries in the world (Åsard, 2016; Alm, 2003). We generally listen to the same type of music, watch the same type of movies, and watch the same type of TV-shows. We don't, however, have similar political systems (as demonstrated by recent events). By contrast, there is arguably no (significant) difference in culture, politics, or economics between the Nordic countries. In fact, there are probably more cultural differences *within* the countries than between.

A relevant question is how to also include minority languages from other language families, such as Sámi. A natural suggestion for this specific case is to consider a Uralic language model, which would include languages such as Finnish, Hungarian, Estonian, as well as the smaller languages Erzya, Moksha, Mari, Udmurt, Sámi, and Komi.

7 Argument 6: Transparency

The largest concurrent language models are not publicly available. Few have probably missed the controversy surrounding the initial decision of Open AI to *not* release GPT-2 due to concerns of adversarial usages.⁹ As we know, GPT-2 was eventually released in full, and there are now GPT-2 models available in many other languages. The original GPT-3 model is however not yet openly available (*Open AI* is beginning to look like a misnomer), but there are several open-source efforts to provide competing, or at least alternative, models.^{10,11}

This lack of transparency obviously limits the ability for other researchers not only to investigate this type of model, but also to contribute to its future development. A collaborative Nordic effort would ensure inclusivity in the development, as well as accessibility to the final model.

⁹openai.com/blog/better-language-models/

¹⁰github.com/EleutherAI/gpt-neo

¹¹github.com/sberbank-ai/ru-gpts

8 Conclusions

Based on the considerations raised in this paper, we argue that we – the Nordic NLP community – **should work together to build a truly large-scale Nordic language model**, for the Nordic languages, by Nordic researchers. We believe that such a resource will be extremely beneficial for Nordic NLP, and that it will have the potential to reduce the environmental impact of continuously training new models.

References

- Martin Alm. 2003. America and the future of sweden: Americanization as controlled modernization. *American Studies in Scandinavia*, 35(2):64–72.
- Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. 2020. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. In *ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems*.
- E. Åsard. 2016. *Det blågula stjärnbaneret: USA:s närvaro och inflytande i Sverige*. Carlssons.
- Michele Banko and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, ACL '01*, page 26–33, USA. Association for Computational Linguistics.
- Emily Bender, Timnit Begru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *FACCT '21*.
- Shikha Bordia and Samuel R. Bowman. 2019. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15, Minneapolis, Minnesota. Association for Computational Linguistics.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. <http://arxiv.org/abs/2005.14165> Language models are few-shot learners.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. <https://doi.org/10.18653/v1/2020.acl-main.747> Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- William Fedus, Barret Zoph, and Noam Shazeer. 2021. <http://arxiv.org/abs/2101.03961> Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity.
- K Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.
- Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.
- Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.
- Chuan Li. 2020. Openai’s gpt-3 language model: A technical overview. <https://lambdalabs.com/blog/demystifying-gpt-3/>. Accessed: 2021-02-05.
- Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.

- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, Open AI.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Magnus Sahlgren and Alessandro Lenci. 2016. The effects of data size and frequency range on distributional semantic models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 975–980, Austin, Texas. Association for Computational Linguistics.
- Or Sharir, Barak Peleg, and Yoav Shoham. 2020. <http://arxiv.org/abs/arXiv:2004.08900> The cost of training nlp models: A concise overview.
- Iben Stampe Sletten, Arne Torp, Kaisa Häkkinen, Mikael Svonni, and Carl Christian Olsen. 2005. *Nordens språk - med rötter och fötter*. Number 2004:008 in Nord. Nordisk ministerråd.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, pages 3266–3280. Curran Associates, Inc.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763. Curran Associates, Inc.

Decentralized Word2Vec Using Gossip Learning^{*}

Abdul Aziz Alkathiri[†] Lodovico Giaretta[†] Šarūnas Girdzijauskas[†] Magnus Sahlgren[‡]

[†] KTH Royal Institute of Technology

[‡] RISE Research Institutes of Sweden

{aatba, lodovico, sarunasg}@kth.se

sahlgren@ri.se

Abstract

Advanced NLP models require huge amounts of data from various domains to produce high-quality representations. It is useful then for a few large public and private organizations to join their corpora during training. However, factors such as legislation and user emphasis on data privacy may prevent centralized orchestration and data sharing among these organizations. Therefore, for this specific scenario, we investigate how gossip learning, a massively-parallel, data-private, decentralized protocol, compares to a shared-dataset solution. We find that the application of Word2Vec in a gossip learning framework is viable. Without any tuning, the results are comparable to a traditional centralized setting, with a reduction in ground-truth similarity scores as low as 4.3%. Furthermore, the results are up to 54.8% better than independent local training.

1 Introduction

Machine learning models, and especially deep learning models (LeCun, 2015) used to represent complex systems, require huge amounts of data. This is also the case with large-scale Natural Language Processing (NLP) models. Moreover, these models benefit from merging various sources of text from different domains to obtain a more complete representation of the language.

For this reason, a small number of separate organizations (for example, government agencies)

may want to train a complex NLP model using the combined data of their corpora to overcome the limitations of each single corpus. However, the typical solution in which all data is moved to a centralized system to perform the training may not be viable, as that could potentially violate privacy laws or data collection agreements and would require all organization to trust the owner of the system with access to their data.

This problem can potentially be solved using massively-parallel, data-private, decentralized approaches – that is, distributed approaches where training is done directly on the machines that produce and hold the data, without having to share or transfer it and without any central coordination – such as gossip learning (Ormándi et al., 2013).

Therefore, we seek to investigate, in the scenario of a small group of large organizations, how models that are produced from the corpus of each node on a decentralized, fully-distributed, data-private configuration, i.e. gossip learning, compare to models trained using a traditional centralized approach where all the data are moved from local machines to a data center. Furthermore, we investigate how these models compare to models trained locally using local data only, without any cooperation.

Our results show that the Word2Vec (Mikolov et al., 2013b) models trained by our implementation of gossip learning are close to models produced by its centralized counterpart setting, in terms of quality of the generated embeddings, and vastly better than what simple local training can produce.

2 Background and related work

The main technique for massively-parallel, data-private training is federated learning (Yang et al., 2019), a centralized approach where each worker node calculates an update of the model based on local data. This gradient is then sent back to the central node which aggregates all these gradients to produce an updated global model which is sent

^{*} This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813162. The content of this paper reflects the views only of their author (s). The European Commission/ Research Executive Agency are not responsible for any use that may be made of the information it contains.

back to the workers. This approach, however, suffers from issues such as the presence of a central node which may act as a privileged “gatekeeper”, as well as reliability issue on the account of that central node.

Unlike centralized approaches, with decentralized machine learning all the nodes in the network execute the same protocols with the same level of privileges, mitigating chances of exploitation by malicious actors. Furthermore, with a peer-to-peer network protocol, decentralized machine learning can virtually scale up to unlimited sizes and be more fault-tolerant, as the network traffic is spread out across multiple links, and not all directed to a single central location. One such approach is the gossip learning protocol (Ormándi et al., 2013).

The gossip communication approach refers to a set of decentralized communication protocols inspired by the behaviour of the spread of gossip socially among people (Shah, 2009). First introduced for the purpose of efficiently synchronizing distributed servers (Demers et al., 1987), it has also applied to various problems, such as data aggregation (Kempe et al., 2003) and failure detection (Van Renesse et al., 1998).

3 Gossip Learning

Gossip learning is an asynchronous, data-parallel, decentralized averaging approach based on gossip communications. It has been shown to be effective when applied to various ML techniques, including binary classification with support vector machines (Ormándi et al., 2013), k-means clustering (Berta and Jelasity, 2017) and low-rank matrix decomposition (Hegedűs et al., 2016). However, these implementations of gossip learning are limited to simple scenarios, where each node holds a single data point and network communications are unrestricted. Giaretta and Girdzijauskas (2019) showed that the gossip protocol can be extended to a wider range of more realistic conditions. However, they identify issues with certain conditions that appear in some real-world scenarios, such as bias towards the data stored with faster communication speeds and the impact of network topologies on the convergence speed of models.

Algorithm 1 shows the general structure of gossip learning as introduced by Ormándi et al. (2013). Intuitively, models perform random walks over the network, merging with each other and training on local data at each node visited.

Algorithm 1: Generic Gossip Learning.

```

 $m_{cur} \leftarrow \text{INITMODEL}()$ 
 $m_{last} \leftarrow m_{cur}$ 
loop
   $\text{WAIT}(\Delta)$ 
   $p \leftarrow \text{RANDOMPEER}()$ 
   $\text{SEND}(p, m_{cur})$ 
end loop
procedure  $\text{ONMODELRECEIVED}(m_{rec})$ 
   $m_{cur} \leftarrow \text{UPDATE}(\text{MERGE}(m_{rec}, m_{last}))$ 
   $m_{last} \leftarrow m_{rec}$ 
end procedure

```

Each node, upon receiving a model from a peer, executes `ONMODELRECEIVED`. The received model m_{rec} and the previous received model m_{last} are averaged weight-by-weight. The resulting model is trained on a single batch of local data and stored as m_{cur} . At regular intervals, m_{cur} is sent to a random peer.

We simulate gossip learning on a single machine, using synchronous iterations. This approximation works well under the assumption that all nodes have similar speeds. If that is not the case, additional measures must be taken to ensure correct model behaviour (Giaretta and Girdzijauskas, 2019).

4 Methodology

While gossip learning could be applied to most NLP algorithms, in this work we use Word2Vec (Mikolov et al., 2013a) because it is simple, small, and fast, thus allowing us to perform larger experiments on limited hardware resources. Additionally, it is a well-known, well-understood technique, allowing us to more easily interpret the results.

The dataset used is the Wikipedia articles dump (Wikimedia Foundation, 2020) of more than 16GB, which contains over 6 million articles and in wiki-text format with embedded XML metadata. From this dump we extract the articles belonging to the following 5 Wikipedia categories of similar size: *science*, *politics*, *business*, *humanities* and *history*.

To measure the quality of the word embeddings produced by a specific model, we collect the $k = 8$ closest words to a target word w_t according to said model. We then assign to each of these words a score based on their *ground-truth* cosine similarity to w_t . We repeat this process for a set of (contextually ambiguous) target words W_t ($|W_t| = 23$) and use the total sum as the quality of the model. We estimate the ground-truth word similarities using a high-quality reference model, more specifically a state of the art Word2Vec model trained on the

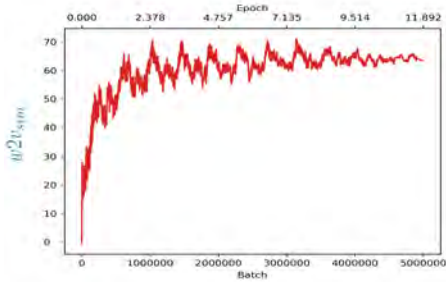


Figure 1. $w2v_{sim}$ evolution for centralized training.

Google News dataset, which uses a similar embedding size ($d = 300$) and contains a vocabulary of 3 million words (Google Code Archive, 2013).

This metric can be defined as

$$w2v_{sim}(M) = \sum_{w_t \in W_t} \sum_{w \in N_M^k(w_t)} sim_R(w, w_t)$$

where M is the model to be evaluated, $N_M^k(\cdot)$ is the top- k neighbourhood function over the embeddings of M and sim_R is the *ground-truth* cosine similarity measure defined based on the reference model.

5 Experimental results

To establish the baseline to compare to, the first experiment is in the traditional non-distributed, centralized configuration of Word2Vec. The baseline $w2v_{sim}$ value is 64.479, as shown in Figure 1.

We simulate gossip learning with 10 nodes, with three different data distributions. In the *r-balanced* distribution, the corpora of the nodes have similar sizes and are randomly drawn from the dataset. In the *r-imbalanced* distribution, the corpora are similarly drawn at random, but have skewed sizes (up to a 4:1 ratio). Finally, in the *topicwise* distribution, the dataset is divided between the nodes based on the 5 Wikipedia categories, with two nodes splitting each category.

The intuition behind dividing the texts by topic is that often times the corpora of organizations are limited to a specific domain. And setting imbalanced content sizes in one of the distributions can provide insights into how the learning is affected when some nodes have significantly bigger corpora than others. Both these configurations are very relevant to the practical applicability of this work, as they both reflect common real-world scenarios.

Exchange frequency	Data distribution	$w2v_{sim}$	$w2v_{sim}$ reduction w.r.t. baseline
Frequent	<i>topicwise</i>	60.606	6.390%
	<i>r-balanced</i>	59.936	7.580%
	<i>r-imbalanced</i>	60.122	7.247%
Infrequent	<i>topicwise</i>	61.840	4.267%
	<i>r-balanced</i>	60.910	5.859%
	<i>r-imbalanced</i>	60.968	5.759%

Table 1. Summary of $w2v_{sim}$ scores for all tested gossip learning configurations.

The formulation of gossip learning presented in Section 3 requires the nodes to exchange their models after every local batch update. As complex NLP models can require millions of training batches, the communication overheads can quickly add up. We thus investigate the effect of reducing the exchange frequency while still maintaining the same number of training batches. More precisely, we repeat the same tests but limit the nodes to exchange the models every 50 batch updates, thus reducing overall communication by a factor of 50.

Figure 2 shows the evolution of the trained models for all combinations of exchange frequency and data distribution. Table 1 summarizes the final scores and compares them to the baseline. In all combinations, the model quality is quite comparable to the traditional centralized configuration. In fact, for the gossip learning with infrequent exchange configuration, there is a slight improvement over the frequent exchange in terms of training time required and $w2v_{sim}$ value. This indicates that the original gossip learning formulation has significant margins of optimization in terms of communication overhead. Furthermore, the relatively unchanged values of $w2v_{sim}$ between the data distributions, in spite of the heterogeneity/homogeneity of the node contents and their sizes, show that gossip learning is robust to topicality and local dataset size. The results suggest that the quality of word embeddings produced using gossip learning is comparable to what can be achieved by training in a traditional centralized configuration using the same parameters, with a loss of quality as low as 4.6% and never higher than 7.7%.

We perform one more experiment, in which each node independently trains a model on its local data only, using the *topicwise* distribution. The $w2v_{sim}$ values do not converge as quickly and range from 41.657 to 56.570 (see Figure 3). This underscores the importance for different organizations to collab-

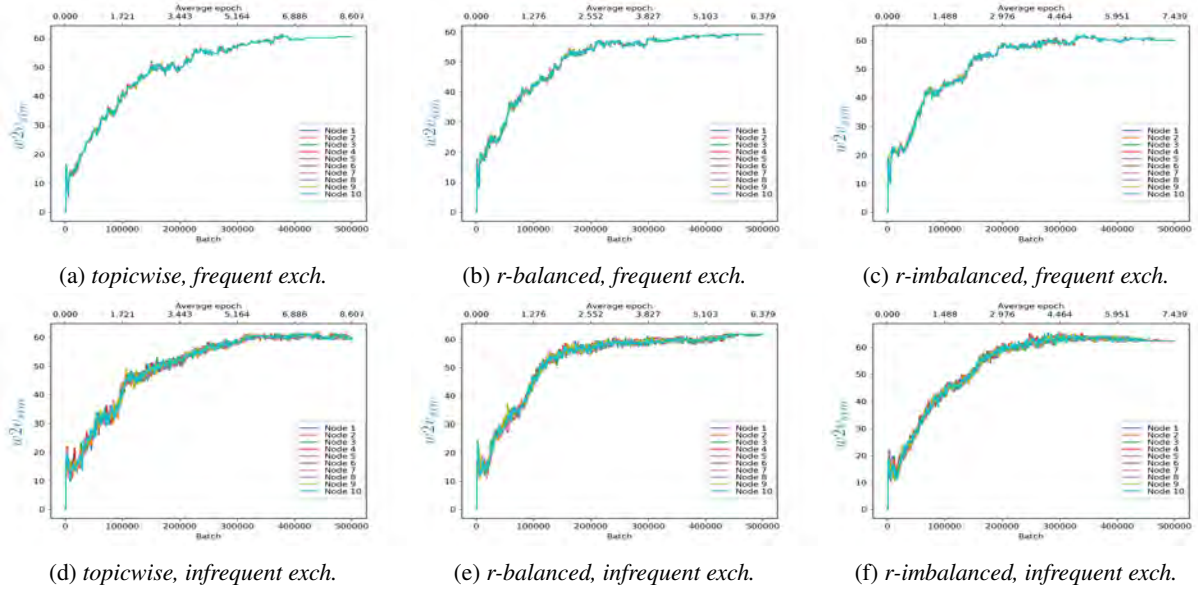


Figure 2. Evolution of $w2v_{sim}$ similarity scores for all tested data distributions and exchange frequencies.

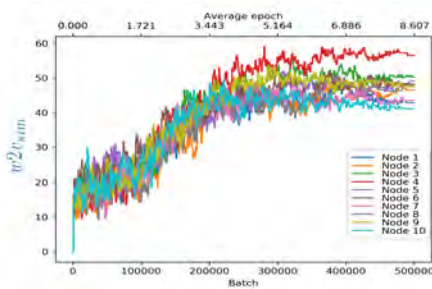


Figure 3. Local, independent training at each node: $w2v_{sim}$ similarity score evolution.

orate to overcome the specificity of local corpora, as this can increase model quality by as much as 54.8%.

6 Limitations and future work

Although the experimental setup of this research takes into account parameters and conditions which simulate real-world scenarios, it is still limited in scope. For instance, the network conditions were assumed to be perfect. Furthermore, security and privacy considerations in the area of networking were not taken into account. Although they were not the focus of this research, their significance cannot be overlooked. Investigating the behaviour of the proposed solution in more realistic network conditions is therefore a possible avenue of research.

A single, simple NLP algorithm (Word2Vec) was evaluated in this work. This is due to the purpose of this research, which was to test the viability of

gossip learning and compare it to a centralized solution in a specific scenario. Evaluating more recent, contextualized NLP models, such as BERT (Devlin et al., 2019) would be an interesting research direction, as these can better capture the different meanings of the same words in multiple domains.

Finally, the experiments were run without extensive hyperparameter optimization. Given the satisfactory results obtained, it is likely that a proper tuning, based on state of the art distributed training research (Shallue et al., 2018), could lead to gossip learning matching or even surpassing the quality of traditional centralized training.

7 Conclusions

Motivated by the scenario where various organizations wish to jointly train a large, high-quality NLP model without disclosing their own sensitive data, the goal of this work was to test whether Word2Vec could be implemented on top of gossip learning, a massively-parallel, decentralized, data-private framework.

The quality of the word embeddings produced using gossip learning is close to what can be achieved in a traditional centralized configuration using the same parameters, with a loss of quality as low as 4.3%, a gap that might be closed with more advance tuning. The frequency of model exchange, which affects bandwidth requirements, has also been reduced 50 times without negative effects. Finally, gossip learning can achieve up to 54.8% better quality than local training alone, motivating

the need for joint training among organizations.

The results of this work therefore show that gossip learning is a viable solution for large-scale, data-private NLP training in real-world applications.

References

- Árpád Berta and Márk Jelasity. 2017. Decentralized management of random walks over a mobile phone network. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 100–107. IEEE.
- Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. 1987. Epidemic algorithms for replicated database maintenance. In *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pages 1–12.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Lodovico Giarretta and Šarūnas Girdzijauskas. 2019. Gossip learning: off the beaten path. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1117–1124. IEEE.
- Google Code Archive. 2013. 3top/word2vec-api.
- István Hegedűs, Árpád Berta, Levente Kocsis, András A Benczúr, and Márk Jelasity. 2016. Robust decentralized low-rank matrix decomposition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(4):1–24.
- David Kempe, Alin Dobra, and Johannes Gehrke. 2003. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 482–491. IEEE.
- Yann LeCun. 2015. Yoshua bengio, and geoffrey hinton. *Deep learning. nature*, 521(7553):436–444.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Róbert Ormándi, István Hegedűs, and Márk Jelasity. 2013. Gossip learning with linear models on fully distributed data. *Concurrency and Computation: Practice and Experience*, 25(4):556–571.
- Devavrat Shah. 2009. Network gossip algorithms. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3673–3676. IEEE.
- Christopher J. Shallue, Jaehoon Lee, Joseph M. Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E. Dahl. 2018. Measuring the effects of data parallelism on neural network training. *CoRR*, abs/1811.03600.
- Robbert Van Renesse, Yaron Minsky, and Mark Hayden. 1998. A gossip-style failure detection service. In *Middleware ’98*, pages 55–70. Springer.
- Wikimedia Foundation. Wikipedia dump at <https://dumps.wikimedia.org/backup-index.html> [online]. 2020.
- Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19.

Multilingual ELMo and the Effects of Corpus Sampling

Vinit Ravishankar, Andrey Kutuzov, Lilja Øvrelid, Erik Velldal

Language Technology Group
Department of Informatics
University of Oslo

{vinitr, andreku, liljao, erikve}@ifi.uio.no

Abstract

Multilingual pretrained language models are rapidly gaining popularity in NLP systems for non-English languages. Most of these models feature an important corpus sampling step in the process of accumulating training data in different languages, to ensure that the signal from better resourced languages does not drown out poorly resourced ones. In this study, we train multiple multilingual recurrent language models, based on the ELMo architecture, and analyse both the effect of varying corpus size ratios on downstream performance, as well as the performance difference between monolingual models for each language, and broader multilingual language models. As part of this effort, we also make these trained models available for public use.

1 Introduction

As part of the recent emphasis on language model pretraining, there also has been considerable focus on multilingual language model pretraining; this is distinguished from merely training language models in multiple languages by the creation of a multilingual space. These have proved to be very useful in ‘zero-shot learning’; i.e., training on a well-resourced language (typically English), and relying on the encoder’s multilingual space to create reasonable priors across languages.

The main motivation of this paper is to study the effect of corpus sampling strategy on downstream performance. Further, we also examine the utility of multilingual models (when constrained to monolingual tasks), over individual monolingual models, one per language. This paper therefore has two main contributions: the first of these is a multilingual ELMo model that we hope would

see further use in probing studies as well as evaluative studies, downstream; we train these models over 13 languages, namely Arabic, Basque, Chinese, English, Finnish, Hebrew, Hindi, Italian, Japanese, Korean, Russian, Swedish and Turkish. The second contribution is an analysis of sampling mechanism on downstream performance; we elaborate on this later.

In Section 2 of this paper, we contextualise our work in the present literature. Section 3 describes our experimental setup and Section 4 our results. Finally, we conclude with a discussion of our results in Section 5.

2 Prior work

Multilingual embedding architectures (static or contextualised) are different from cross-lingual ones (Ruder et al., 2019; Liu et al., 2019) in that they are not products of aligning several monolingual models. Instead, a deep neural model is trained end to end on texts in multiple languages, thus making the whole process more straightforward and yielding truly multilingual representations (Pires et al., 2019). Following Artetxe et al. (2020), we will use the term ‘deep multilingual pretraining’ for such approaches.

One of the early examples of deep multilingual pretraining was BERT, which featured a multilingual variant trained on the 104 largest language-specific Wikipedias (Devlin et al., 2019). To counter the effects of some languages having overwhelmingly larger Wikipedias than others, Devlin et al. (2019) used exponentially smoothed data weighting; i.e., they exponentiated the probability of a token being in a certain language by a certain α , and re-normalised. This has the effect of ‘squashing’ the distribution of languages in their training data; larger languages become smaller, to avoid drowning out the signal from smaller languages. One can also look at this technique as a sort of sampling. Other multilingual models,

such as XLM (Lample and Conneau, 2019) and its larger variant, XLM-R (Conneau et al., 2020), use different values of α for this sampling (0.5 and 0.3 respectively). The current paper is aimed at analysing the effects of different α choices; in spirit, this work is very similar to Arivazhagan et al. (2019); where it differs is our analysis on downstream tasks, as opposed to machine translation, where models are trained and evaluated on a very specific task. We also position our work as a resource, and we make our multilingual ELMo models available for public use.

3 Experimental setup

3.1 Background

When taken to its logical extreme, sampling essentially reduces to truncation, where all languages have the same amount of data; thus, in theory, in a truncated model, no language ought to dominate any other. Of course, for much larger models, like the 104-language BERT, this is unfeasible, as the smallest languages are too small to create meaningful models. By selecting a set of languages such that the smallest language is still reasonably sized for the language model being trained, however, we hope to experimentally determine whether truncation leads to truly neutral, equally capable multilingual spaces; if not, we attempt to answer the question of whether compression helps at all.

Our encoder of choice for this analysis is an LSTM-based ELMo architecture introduced by Peters et al. (2018). This might strike some as a curious choice of model, given the (now) much wider use of transformer-based architectures. There are several factors that make ELMo more suitable for our analysis. Our main motivation was, of course, resources – ELMo is far cheaper to train, computationally. Next, while pre-trained ELMo models already exist for several languages (Che et al., 2018; Ulčar and Robnik-Šikonja, 2020), there is, to the best of our knowledge, no multilingual ELMo. The release of our multilingual model may therefore also prove to be useful in the domain of probing, encouraging research on multilingual encoders, constrained to recurrent encoders.

3.2 Sampling

Our initial starting point for collecting the language model training corpora were the CoNLL

2017 Wikipedia/Common Crawl dumps released as part of the shared task on Universal Dependencies parsing (Ginter et al., 2017); we extracted the Wikipedia portions of these corpora for our set of 13 languages. This gives us a set of fairly typologically distinct languages, that still are not entirely poorly resourced. The smallest language in this collection, Hindi, has ~ 91 M tokens, which we deemed sufficient to train a reasonable ELMo model.

Despite eliminating Common Crawl data, this gave us, for our set of languages, a total corpus size of approximately 35B tokens, which would be an unfeasible amount of data given computational constraints. We therefore selected a baseline model to be somewhat synthetic – note that this is a perfectly valid choice given our goals, which were to compare various sampling exponents. Our ‘default’ model, therefore, was trained on data that we obtained by weighting this ‘real-world’ Wikipedia data. The largest α we could use, that would still allow for feasible training, was $\alpha = 0.4$ (further on, we refer to this model as M0.4); this gave us a total corpus size of ~ 4 B tokens. Our second, relatively more compressed model, used $\alpha = 0.2$ (further on, M0.2); giving us a total corpus size of ~ 2 B tokens; for our final, most compressed model (further on, TRUNC), we merely truncated each corpus to the size of our smallest corpus (Hindi; 91M), giving us a corpus sized ~ 1.2 B tokens. Sampling was carried out as follows: if the probability of a token being sampled from a certain language i is p_i , the adjusted probability is given by $q_i = \frac{p_i}{\sum_{j=1}^N p_j}$. Note that this is a similar sampling strategy to the one followed by more popular models, like mBERT. We trained an out-of-the box ELMo encoder for approximately the same number of steps on each corpus; this was equivalent to 2 epochs for M0.4 and 3 for M0.2.

Detailed training hyperparameters and precise corpus sizes are presented in Appendices A and B.

3.3 Tasks

While there is a dizzying array of downstream evaluation tasks for monolingual models, looking to evaluate multilingual models is a bit harder. We settled on a range of tasks in two different groups:

1. **Monolingual tasks:** these tasks directly test the monolingual capabilities of the model, per language. We include PoS tagging and

dependency parsing in this category. In addition to our multilingual models, we also evaluate our monolingual ELMo variants on these tasks.

2. **Transfer tasks:** these tasks involve leveraging the model’s multilingual space, to transfer knowledge from the language it was trained on, to the language it is being evaluated on. These tasks include natural language inference and text retrieval; we also convert PoS tagging into a transfer task, by training our model on English and asking it to tag text in other languages.

In an attempt to illuminate precisely what the contribution of the different ELMo models is, we ensure that our decoder architectures – that translate from ELMo’s representations to the task’s label space – are kept relatively simple, particularly for lower-level tasks. We freeze ELMo’s parameters: this is not a study on fine-tuning.

The tasks that we select are a subset of the tasks mentioned in XTREME (Hu et al., 2020); i.e., the subset most suitable to the languages we trained our encoder on. A brief description follows:

PoS tagging: For part-of-speech tagging, we use Universal Dependencies part-of-speech tagged corpora (Nivre et al., 2020). Built on top of our ELMo-encoder is a simple MLP, that maps representations onto the PoS label space.

PoS tagging (transfer): We use the same architecture as for regular PoS tagging, but train on English and evaluate on our target languages.

Dependency parsing: We use dependency-annotated Universal Dependencies corpora; our metrics are both unlabelled and labelled attachment scores (UAS/LAS). Our parsing architecture is a biaffine graph-based parser (Dozat and Manning, 2018).

XNLI: A transfer-based language inference task; we use Chen et al.’s 2017 ESIM architecture, train a tagging head on English, and evaluate on the translated dev portions of other languages (Conneau et al., 2018).

Tatoeba: The task here is to pick out, for each sentence in our source corpus (English), the appropriate translation of the sentence in our target language corpus. This, in a sense, is the most ‘raw’ tasks; target language sentences are

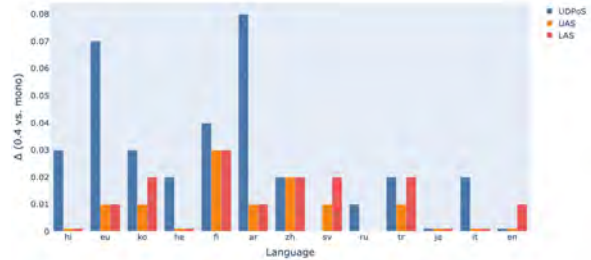


Figure 1: Performance difference between monolingual and multilingual models, on our monolingual tasks. Absent bars indicate that the language was missing.

ranked based on similarity. We follow Hu et al. (2020) and use the Tatoeba dataset.

We tokenize all our text using the relevant UD-Pipe (Straka et al., 2019) model, and train/evaluate on each task three times; the scores we report are mean scores.

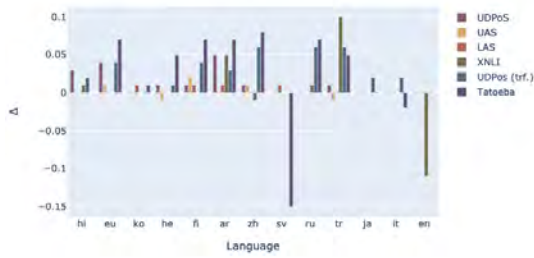
4 Results

First, we examine the costs of multilingualism, as far as monolingual tasks are concerned. We present our results on our monolingual tasks in Figure 1. Monolingual models appear to perform consistently better, particularly PoS tagging; this appears to be especially true for our under-resourced languages, strengthening the claim that compression is necessary to avoid drowning out signal. For PoS tagging, the correlation between performance difference (monolingual vs. M0.4) and corpus size is highly significant ($\rho = 0.74$; $p = 0.006$).

	PoS	UAS	LAS	PoS (trf.)	XNLI	Tatoeba
MONO	0.86	0.86	0.81	-	-	-
M0.4	0.83	0.85	0.80	0.36	0.45	0.18
M0.2	0.84	0.85	0.80	0.39	0.46	0.21
TRUNC	0.83	0.85	0.80	0.36	0.45	0.13

Table 1: Average scores for each task and encoder; non-monolingual best scores in bold.

We find that compression appears to result in visible improvements, when moving from $\alpha = 0.4$ to $\alpha = 0.2$. These improvements, while not dramatic, apply across the board (see Table 1), over virtually all task/language combinations; this is visible in Figure 2a. Note the drop in performance on certain tasks for English, Swedish and Italian –



(a) M0.2 vs. M0.4



(b) TRUNC vs. M0.4

Figure 2: Performance differences between our models on our selected tasks.

we hypothesise that this is due to Swedish and Italian being closer to English (our most-sampled language), and therefore suffering from the combination of the drop in their corpus sizes, as well as the more significant drop in English corpus size. The Pearson correlation between the trend in performance for PoS tagging and the size of a language’s corpus is statistically significant ($\rho = 0.65$; $p = 0.02$); note that while this is over multiple points, it is single runs per data point.

Figure 2b also shows the difference in performance between the truncated model, TRUNC, and M0.4; this is a lot less convincing than the difference to M0.2, indicating that no additional advantage is to be gained by downsampling data for better-resourced languages.

We include full, detailed results in Appendix C.

Cross-lingual differences Finally, in an attempt to study the differences in model performance across languages, we examine the results of all models on Tatoeba. This task has numerous advantages for a more detailed analysis; i) it covers all our languages, bar Hindi, ii) the results have significant variance across languages, and iii) the task does not involve any additional training. We present these results in Figure 3.

We observe that M0.2 consistently appears to perform better, as illustrated earlier. Performance does not appear to have much correlation with corpus size; however, the languages for which M0.4 performs better are Swedish and Italian, coincidentally, the only other Latin-scripted Indo-European languages. Given the specific nature of Tatoeba, which involves picking out appropriate translations, these results make more sense: these languages receive not only the advantage of having more data for themselves, but also from the

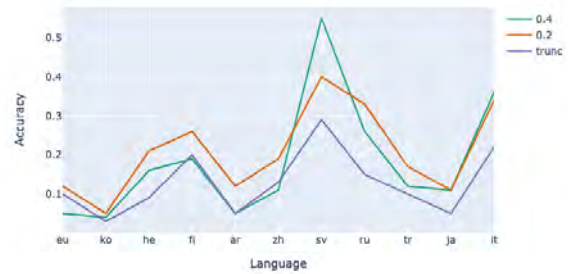


Figure 3: Accuracy on Tatoeba per model

additional data available to English, which in turn optimises their biases solely by virtue of language similarity.

5 Discussion

Our results allow us to draw conclusions that come across as very ‘safe’: some compression helps, too much hurts; when compression does help, however, the margin appears rather moderate yet significant for most tasks, even given fewer training cycles. Immediately visible differences along linguistic lines do not emerge when ratios differ, despite the relative linguistic diversity of our language choices; we defer analysis of this to a future work, that is less focused on downstream analysis, and more on carefully designed probes that might illuminate the difference between our models’ internal spaces. Note that a possible confounding factor in our results is also the complexity of the architectures we build on top of mELMO: they also have significant learning capacity, and it is not implausible that whatever differences there are between our models, are drowned out by highly parameterised downstream decoders.

To reiterate, this study is not (nor does it aim to be) a replication of models with far larger parameter spaces and more training data. This is something of a middle-of-the-road approach; future work could involve this sort of evaluation on downscaled transformer models, which we shy away from in order to provide a usable model release. We hope that the differences between these models provide some insight, and pave the way for further research, not only specifically addressing the question of sampling from a perspective of performance, but also analytically. There has already been considerable work in this direction on multilingual variants of BERT (Pires et al., 2019; Chi et al., 2020), and we hope that this work motivates papers applying the same to recurrent mELMo, as well as comparing and contrasting the two. The ELMo models described in this paper are publicly released via NLPL Vector Repository.¹

Acknowledgements

Our experiments were run on resources provided by UNINETT Sigma2 - the National Infrastructure for High Performance Computing and Data Storage in Norway, under the NeIC-NLPL umbrella.

References

- Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges. *arXiv:1907.05019 [cs]*. ArXiv: 1907.05019.
- Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. 2020. A call for more rigor in unsupervised cross-lingual learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7375–7388, Online. Association for Computational Linguistics.
- Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.
- Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668. ArXiv: 1609.06038.
- Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.
- Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. CoNLL 2017 shared task - automatically annotated raw texts and word embeddings. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *arXiv preprint arXiv:2003.11080*.
- Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

¹<http://vectors.nlpl.eu/repository/>

Qianchu Liu, Diana McCarthy, Ivan Vulić, and Anna Korhonen. 2019. Investigating cross-lingual alignment methods for contextualized embeddings with token-level evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 33–43, Hong Kong, China. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Milan Straka, Jana Straková, and Jan Hajic. 2019. UD-Pipe at SIGMORPHON 2019: Contextualized embeddings, regularization with morphological categories, corpora merging. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 95–103, Florence, Italy. Association for Computational Linguistics.

Matej Ulčar and Marko Robnik-Šikonja. 2020. High quality ELMo embeddings for seven less-resourced languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4731–4738, Marseille, France. European Language Resources Association.

A Hyperparameters

B Corpus sizes

C Detailed results

Param	Value
Layers	2
Output dimensionality	2048
Batch size	192
Negative samples per batch	4096
Vocabulary size	100,000
Number of epochs	2 (M0.4); 3 (M0.2)

Table 2: Models were bidirectional LSTMs. Monolingual models were trained on individual sizes given at $\alpha = 0.4$.

Language	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH	Total
M0.4	242.29	585.52	113.42	239.57	208.46	91.74	468.45	460.53	184.63	379.9	366.86	396.01	282.76	4020.14
M0.2	149.09	231.76	102.01	148.25	138.29	91.74	207.3	205.54	130.15	186.68	183.45	190.6	161.06	2125.92
TRUNC	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	91.74	1192.62

Table 3: Corpus sizes, in million tokens

Language		AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
POS	MONO	0.89	0.89	0.88	0.82	0.84	0.9	0.91	0.94	0.67	0.88	-	0.83	0.86
	0.4	0.81	0.89	0.81	0.78	0.82	0.87	0.89	0.94	0.64	0.87	-	0.81	0.84
	0.2	0.86	0.89	0.85	0.79	0.83	0.9	0.89	0.94	0.64	0.87	-	0.82	0.85
	TRUNC	0.82	0.89	0.84	0.8	0.82	0.9	0.88	0.93	0.63	0.86	-	0.81	0.85
UAS	MONO	0.86	0.89	0.84	0.88	0.89	0.94	0.93	0.95	0.8	-	0.85	0.69	0.8
	M0.4	0.85	0.89	0.83	0.85	0.89	0.94	0.93	0.95	0.79	-	0.84	0.68	0.78
	M0.2	0.85	0.89	0.84	0.87	0.88	0.94	0.93	0.95	0.79	-	0.84	0.67	0.79
	TRUNC	0.85	0.89	0.83	0.86	0.89	0.94	0.93	0.95	0.78	-	0.84	0.68	0.79
LAS	MONO	0.79	0.86	0.79	0.84	0.84	0.9	0.9	0.94	0.74	-	0.81	0.59	0.74
	0.4	0.78	0.85	0.78	0.81	0.84	0.9	0.9	0.94	0.72	-	0.79	0.57	0.72
	0.2	0.79	0.85	0.78	0.82	0.84	0.9	0.9	0.94	0.73	-	0.8	0.57	0.72
	TRUNC	0.79	0.85	0.78	0.82	0.84	0.9	0.9	0.93	0.72	-	0.79	0.57	0.72
POS (trf.)	0.4	0.23	0.89	0.25	0.43	0.36	0.31	0.52	0.22	0.18	0.49	-	0.23	0.22
	0.2	0.26	0.89	0.29	0.47	0.37	0.33	0.54	0.24	0.18	0.55	-	0.29	0.28
	TRUNC	0.23	0.89	0.3	0.48	0.32	0.26	0.48	0.2	0.17	0.49	-	0.27	0.28
XNLI	M0.4	0.41	0.67	-	-	-	0.44	-	-	-	0.48	-	0.35	0.35
	M0.2	0.46	0.56	-	-	-	0.45	-	-	-	0.49	-	0.45	0.34
	TRUNC	0.43	0.66	-	-	-	0.43	-	-	-	0.43	-	0.43	0.35
Tatoeba	0.4	0.05	-	0.05	0.19	0.16	-	0.36	0.11	0.04	0.26	0.55	0.12	0.11
	0.2	0.12	-	0.12	0.26	0.21	-	0.34	0.11	0.05	0.33	0.4	0.17	0.19
	TRUNC	0.05	-	0.1	0.2	0.09	-	0.22	0.05	0.03	0.15	0.29	0.1	0.13

Table 4: Full score table across all languages, tasks and models

Should we Stop Training More Monolingual Models, and Simply Use Machine Translation Instead?

Tim Isbister
Peltarion

tim.isbister@peltarion.com

Fredrik Carlsson
RISE

fredrik.carlsson@ri.se

Magnus Sahlgren
RISE

magnus.sahlgren@ri.se

Abstract

Most work in NLP makes the assumption that it is desirable to develop solutions in the native language in question. There is consequently a strong trend towards building native language models even for low-resource languages. This paper questions this development, and explores the idea of simply translating the data into English, thereby enabling the use of pretrained, and large-scale, English language models. We demonstrate empirically that a large English language model coupled with modern machine translation outperforms native language models in most Scandinavian languages. The exception to this is Finnish, which we assume is due to inferior translation quality. Our results suggest that machine translation is a mature technology, which raises a serious counter-argument for training native language models for low-resource languages. This paper therefore strives to make a provocative but important point. As English language models are improving at an unprecedented pace, which in turn improves machine translation, it is from an empirical and environmental stand-point more effective to translate data from low-resource languages into English, than to build language models for such languages.

1 Introduction

Although the Transformer architecture for deep learning was only recently introduced (Vaswani et al., 2017), it has had a profound impact on the development in Natural Language Processing (NLP) during the last couple of years. Starting with the seminal BERT model (Devlin et al., 2019), we have witnessed an unprecedented development of new

model variations (Yang et al., 2019; Clark et al., 2020; Raffel et al., 2020; Radford et al., 2019; Brown et al., 2020) with new State Of The Art (SOTA) results being produced in all types of NLP benchmarks (Wang et al., 2018, 2019; Nie et al., 2020).

The leading models are large both with respect to the number of parameters and the size of the training data used to build the model; this correlation between size and performance has been demonstrated by Kaplan et al. (2020). The ongoing scale race has culminated in the 175-billion parameter model GPT-3, which was trained on some 45TB of data summing to around 500 billion tokens (Brown et al., 2020).¹ Turning to the Scandinavian languages, there are no such truly large-scale models available. At the time of writing, there are around 300 Scandinavian models available in the Hugging Face Transformers model repository.² Most of these are translation models, but there is already a significant number of monolingual models available in the Scandinavian languages.³

However, none of these Scandinavian language models are even close to the currently leading English models in parameter size or training data used. As such, we can expect that their relative performance in comparison with the leading English models is significantly worse. Furthermore, we can expect that the number of monolingual Scandinavian models will continue to grow at an exponential pace during the near future. The question is: do we need all these models? Or even: do we need *any* of these models? Can't we simply translate our data and tasks to English and use some suitable English SOTA model to solve the problem? This paper provides an empirical study of this idea.

¹The currently largest English model contains 1.6 trillion parameters (Fedus et al., 2021).

²huggingface.co/models

³At the time of submission, there are 17 monolingual Swedish models available.

Language	Vocab size	Lexical richness	Avg. word length	Avg. sentence length
Swedish	31,478	0.07	4.39	14.75
Norwegian	26,168	0.06	4.21	14.10
Danish	42,358	0.06	4.17	19.55
Finnish	34,729	0.14	5.84	10.69
English	27,610	0.04	3.99	16.87

Table 1: The vocabulary size, Lexical richness, average word length and average sentence length for the Trustpilot sentiment data of each language.

2 Related work

There is already a large, and rapidly growing, literature on the use of multilingual models (Conneau et al., 2020a; Xue et al., 2020), and on the possibility to achieve cross-lingual transfer in multilingual language models (Ruder et al., 2019; Artetxe et al., 2020; Lauscher et al., 2020; Conneau et al., 2020b; Karthikeyan et al., 2020; Nooralahzadeh et al., 2020). From this literature, we know among other things that multilingual models tend to be competitive in comparison with monolingual ones, and that especially languages with smaller amounts of training data available can benefit significantly from transfer effects from related languages with more training data available. This line of study focuses on the possibility to transfer *models* to a new language, and thereby facilitating the application of the model to data in the original language.

By contrast, our interest is to transfer the *data* to another language, thereby enabling the use of SOTA models to solve whatever task we are interested in. We are only aware of one previous study in this direction: Duh et al. (2011) performs cross-lingual machine translation using outdated methods, resulting in the claim that even if perfect translation would be possible, we will still see degradation of performance. In this paper, we use modern machine translation methods, and demonstrate empirically that no degradation of performance is observable when using large SOTA models.

3 Data

In order to be able to use comparable data in the languages under consideration (Swedish, Danish, Norwegian, and Finnish), we contribute a Scandinavian sentiment corpus (ScandiSent),⁴ consisting of data downloaded from `trustpilot.com`. For each language, the corresponding subdomain was used

⁴<https://github.com/timpal01/ScandiSent>

to gather reviews with an associated text. This data covers a wide range of topics and are divided into 22 different categories, such as electronics, sports, travel, food, health etc. The reviews are evenly distributed among all categories for each language.

All reviews have a corresponding rating in the range 1 – 5. The review ratings were polarised into binary labels, and the reviews which received neutral rating were discarded. Ratings with 4 or 5 thus corresponds to a positive label, and 1 or 2 correspond to a negative label.

To further improve the quality of the data, we apply fastText’s language identification model (Joulin et al., 2016) to filter out any reviews containing incorrect language. This results in a balanced set of 10,000 texts for each language, with 7,500 samples for training and 2,500 for testing. Table 1 summarizes statistics for the various datasets of each respective language.

3.1 Translation

For all the Nordic languages we generate a corresponding English dataset by direct Machine Translation, using the Neural Machine Translation (NMT) model provided by Google.⁵ To justifiably isolate the effects of modern day machine translation, we restrict the translation to be executed in prior to all experiments. This means that all translation is executed prior to any fine-tuning, and that the translation model is not updated during training.

4 Models

In order to fairly select a representative pre-trained model for each considered Scandinavian language, we opt for the most popular native model according to Hugging Face. For each considered language, this corresponds to a BERT-Base model, hence each language is represented by a Language Model

⁵<https://cloud.google.com/translate/docs/advanced/translating-text-v3>

Model name in Hugging Face	Language	Data size
KB/bert-base-swedish-cased	sv	3B tokens
TurkuNLP/bert-base-finnish-cased-v1	fi	3B tokens
ltgoslo/norbert	no	2B tokens
DJSammy/bert-base-danish-uncased.BotXO, ai	da	1.6B tokens
bert-base-cased	en	3.3B tokens
bert-base-cased-large	en	3.3B tokens
xlm-roberta-large	multi	295B tokens

Table 2: Models used in the experiments and the size of their corresponding training data. 'B' is short for billion.

Model	sv	no	da	fi	en
BERT-sv	<u>96.76</u>	89.32	90.68	83.40	86.76
BERT-no	90.40	<u>95.00</u>	92.52	83.16	78.52
BERT-da	86.24	89.16	<u>94.72</u>	80.16	85.28
BERT-fi	90.24	86.36	87.72	95.72	84.32
BERT-en	85.72	87.60	87.72	84.16	96.08
BERT-en-Large	91.16	91.88	92.40	89.56	97.00
Translated Into English					
BERT-sv	88.24	87.80	89.68	83.60	-
BERT-no	88.40	86.80	88.44	80.72	-
BERT-da	88.24	84.20	89.12	83.32	-
BERT-fi	90.04	90.08	89.36	86.04	-
BERT-en	95.76	95.48	95.96	92.96	-
BERT-en-Large	97.16	96.56	97.48	94.84	-

Table 3: Accuracy for monolingual models for the native sentiment data (upper part) and machine translated data (lower part). Underlined results are the best results per language in using the native data, while boldface marks the best results considering both native and machine translated data.

Model	sv	no	da	fi	en
XLM-R-large	97.48	97.16	97.68	95.60	97.76
Translated Into English					
XLM-R-large	97.04	96.84	98.24	95.48	-

Table 4: Accuracy on the various sentiment datasets using XLM-R-Large

of identical architecture. The difference between these models is therefore mainly in the quantity and type of texts used during training, in addition to potential differences in training hyperparameters.

We compare these Scandinavian models against the English BERT-Base and BERT-Large models by Google. English BERT-Base is thus identical in architecture to the Scandinavian models, while BERT-Large is twice as deep and contains more than three times the amount of parameters as BERT-Base. Finally, we include XLM-R-Large, in order to compare with a model trained on significantly larger (and multilingual) training corpora.

Table 2 lists both the Scandinavian and English models, together with the size of each models corresponding training corpus.

5 Experiments

5.1 Setup

We fine-tune and evaluate each model towards each of the different sentiment datasets, using the hyperparameters listed in Appendix 5. From this we report the binary accuracy, with the results for the BERT models available in Table 3, and the XLM-R results in Table 4.

5.2 Monolingual Results

The upper part of Table 3 shows the results using the original monolingual data. From this we note a clear diagonal (marked by underline), where the native models perform best in their own respective language. Bert-Large significantly outperforms BERT-Base for all non-English datasets, and it also performs slightly better on the original English data.

Comparing these results with the amount of training data for each model (Table 1), we see a correlation between performance and amount of pre-training data. The Swedish, Finnish and English models have been trained on the most amount of data, leading to slightly higher performance in their native languages. The Danish model which has been trained on the least amount of data, performs the worst on its own native language.

For the cross-lingual evaluation, BERT-Large clearly outperforms all other non-native models. The Swedish model reaches higher performance on Norwegian and Finnish compared to the other non-native Scandinavian models. However, the Norwegian model performs best of the non-native models on the Danish data. Finally, we observe an interesting anomaly in the results on the English data, where the Norwegian model performs considerably worse than the other Scandinavian models.

5.3 Translation Results

The results for the machine translated data, available as the lower part of Table 3, show that BERT-Large outperforms all native models on their native data, with the exception of Finnish. The English BERT-Base reaches higher performance on the machine translated data than the Norwegian and Danish models on their respective native data. The difference between English BERT-Base using the machine translated data, and the Swedish BERT using native data is about 1% unit.

As expected, all Scandinavian models perform significantly worse on their respective machine translated data. We find no clear trend among the Scandinavian models when evaluated on translated data from other languages. But we note that the Danish model performs better on the machine translated Swedish data than on the original Swedish data, and the Finnish model also improves its performance on the other translated data sets (except for Swedish). All models (except, of course, the Finnish model) perform better on the machine trans-

lated Finnish data.

Finally, 4 shows the results from XLM-R-Large, which has been trained on data several orders of magnitude larger than the other models. XLM-R-Large achieves top scores on the sentiment data for all languages except for Finnish. We note that XLM-R produces slightly better results on the native data for Swedish, Norwegian and Finnish, while the best result for Danish is produced on the machine translated data.

6 Discussion & Conclusion

Our experiments demonstrate that it is possible to reach better performance in a sentiment analysis task by translating the data into English and using a large pre-trained English language model, compared to using data in the original language and a smaller native language model. Whether this result holds for other tasks as well remains to be shown, but we see no theoretical reasons for why it would not hold. We also find a strong correlation between the quantity of pre-training data and downstream performance. We note that XLM-R in particular performs well, which may be due to data size, and potentially the ability of the model to take advantage of transfer effects between languages.

An interesting exception in our results is the Finnish data, which is the only task for which the native model performs best, despite XLM-R reportedly having been trained on more Finnish data than the native Finnish BERT model (Conneau et al., 2020a). One hypothesis for this behavior can be that the alleged transfer effects in XLM-R hold primarily for typologically similar languages, and that the performance on typologically unique languages, such as Finnish, may actually be negatively affected by the transfer. The relatively bad performance of BERT-Large on the translated Finnish data is likely due to insufficient quality of the machine translation.

The proposed approach is thus obviously dependent on the existence of a high-quality machine translation solution. The Scandinavian languages are typologically very similar both to each other and to English, which probably explains the good performance of the proposed approach even when using a generic translation API. For other languages, such as Finnish in our case, one would probably need to be more careful in selecting a suitable translation model. Whether the suggested methodology will be applicable to other language

pairs thus depends on the quality of the translations and on the availability of large-scale language models in the target language.

Our results can be seen as evidence for the maturity of machine translation. Even using a generic translation API, we can leverage the existence of large-scale English language models to improve the performance in comparison with building a solution in the native language. This raises a serious counter-argument for the habitual practice in applied NLP to develop native solutions to practical problems. Hence, we conclude with the somewhat provocative claim that it might be unnecessary from an empirical standpoint to train models in languages where:

1. there exists high-quality machine translation models to English,
2. there does not exist as much training data to build a language model.

In such cases, we may be better off relying on existing large-scale English models. This is a clear case for practical applications, where it would be beneficial to only host one large English model and translate all various incoming requests from different languages.

References

- Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. <http://arxiv.org/abs/2005.14165> Language models are few-shot learners.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. <https://doi.org/10.18653/v1/2020.acl-main.747> Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Kevin Duh, Akinori Fujino, and Masaaki Nagata. 2011. Is machine translation ripe for cross-lingual sentiment classification? In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2, HLT '11*, page 429–433, USA. Association for Computational Linguistics.
- William Fedus, Barret Zoph, and Noam Shazeer. 2021. <http://arxiv.org/abs/2101.03961> Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. <http://arxiv.org/abs/2001.08361> Scaling laws for neural language models.
- K Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.
- Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Farhad Nooralahzadeh, Giannis Bekoulis, Johannes Bjerva, and Isabelle Augenstein. 2020. Zero-Shot Cross-Lingual Transfer with Meta Learning. In *Proceedings of EMNLP*. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, Open AI.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, pages 3266–3280. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. <http://arxiv.org/abs/2010.11934> mT5: A massively multilingual pre-trained text-to-text transformer. ArXiv:2010.11934.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for

language understanding. In *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763. Curran Associates, Inc.

A Training Details

Parameters	Value
train_epochs	2
early_stopping	false
optimizer	AdamW
learning_rate	4e-5
batch_size	512
max_seq_length	128
max_grad_norm	1.0

Table 5: Training hyperparameters for the sentiment classification experiments.

Error Analysis of using BART for Multi-Document Summarization: A Study for English and German Language

Timo Johner¹, Abhik Jana¹, and Chris Biemann¹

¹ Language Technology Group, Dept. of Informatics, Universität Hamburg, Germany
me@timojohner.de, jana@informatik.uni-hamburg.de
biemann@informatik.uni-hamburg.de

Abstract

Recent research using pre-trained language models for multi-document summarization tasks lacks a deep investigation of potential erroneous cases and their possible application in other languages. In this work, we apply a pre-trained language model (BART) for multi-document summarization (MDS) task, both with fine-tuning and without fine-tuning. We use two English datasets and one German dataset for this study. First, we reproduce the multi-document summaries for the English language by following one of the recent studies. Next, we show the applicability of the model to the German language by achieving state-of-the-art performance on German MDS. We perform an in-depth error analysis of the followed approach for both languages, which leads us to identify the most notable errors, from made-up facts to topic delimitation. Lastly, we quantify the amount of extractiveness.

1 Introduction

Nowadays, we are confronted with an enormous amount of information through news, mails, social media, etc., which are difficult to absorb for a human being in one go. Hence, there is a pressing need to compress and comprehend this information. Capturing salient details from multiple sources to produce an abridged version is described as Multi-Document Summarization (MDS) (Nenkova and McKeown, 2011) and can be carried out in both an abstractive or extractive manner. MDS has recently become one of the most interesting research topics in the field of Natural Language Processing (NLP). As per the literature, whilst the state-of-the-art models (Gehrmann et al., 2018; Liu et al., 2018) heavily rely on large datasets, recent advances with pre-trained language model systems (Ziegler et al., 2019; Raffel et al., 2020; Lewis et al., 2020) have

shown great potential for the summarization task. While there have been studies to gradually improve the performance of MDS for the English language, MDS for other languages has rarely been attempted. There has also been a lack of in-depth error analysis for the MDS task. In this study, we attempt to analyze and address these issues.

Our main contributions are the following: Firstly, we reproduce recent pre-trained and fine-tuned results for multi-document summarization with the BART model, introduced by Lewis et al. (2020), on two English datasets. Further, we adapt the model for the German language and achieve state-of-the-art performance for the German MDS task, beating the most competitive baseline by a margin of 3.48-8.67%. Secondly, we perform an analysis on the erroneous cases for both languages where we point out general errors and cross-lingual error similarities regarding factfulness and topic delimitation. Additionally, we also investigate the extractiveness of the generated summaries.

2 Related Work

Early approaches on extractive MDS apply term frequency-inverse document frequency (TF-IDF) (McKeown et al., 1999; Goldstein et al., 2000; Radev et al., 2000). Later, Conroy et al. (2006) and Shen and Li (2010), attempt the MDS task with a topic and set-based methodology, respectively. Initial attempts for abstractive multi-document summarization are made by McKeown and Radev (1995) and Radev and McKeown (1998). Barzilay and McKeown (2005) use sentence-fusion for text generation to create summaries across different documents. Haghighi and Vanderwende (2009) build a model based on word frequency and Latent Dirichlet Allocation (LDA) for MDS whereas phrase selection and merging approaches have also been tried (Bing et al., 2015) for the same.

In recent years, neural network architecture is being adapted for several NLP tasks, especially

with the approach of using encoder-decoder architecture. Here, relevant work includes Rush et al. (2015), who propose an attention model for combining extractive and abstractive methods, which is supplemented with document-wide contextualization by Cheng and Lapata (2016) and Nallapati et al. (2016). In a different direction, several graph-based approaches are explored as well (Tan et al., 2017; Yasunaga et al., 2017). Liu et al. (2018) show the feasibility of using Wikipedia as an MDS dataset whereas Fabbri et al. (2019) apply a pointer-generator network with a transformer model complemented with Maximal Marginal Relevance (MMR). Li et al. (2020) explores graph representation and proposes to leverage graphs for abstractive MDS.

Most recently, fine-tuning pre-trained language models have gained a lot of attention for NLP tasks. For summarization, one such work by Raffel et al. (2020) attempted to explore fine-tuning, whereas, in another work, Liu and Lapata (2019) fine-tune BERT for summarization. Later, Hokamp et al. (2020) adapt and fine-tune BART on MDS. Approaches regarding a systematic error analysis of those models were introduced by Huang et al. (2020) who compared BART to other abstractive and extractive methods.

In another direction, attempts have also been made for single-document summarization for non-English text. For instance, single-document summarization of text in German language was done by Parida and Motlicek (2019) who utilized transformer models for abstractive summarization on two datasets — SwissText 2019¹ and Common Crawl². Evaluation of summarization models to non-English data was done by Tauchmann and Mieskes (2020) who applied an automatic evaluation paradigm on the German heterogeneous dataset DBS (Benikova et al., 2016). Since our main focus is on multi-document summarization, we do not explore the literature of single-document summarization extensively.

3 Datasets

For our experiments we use three datasets that exhibit extractive characteristics: two English datasets — CNN/DM (Hermann et al., 2015), Multi-News (Fabbri et al., 2019) and one German dataset — auto-hMDS (Zopf, 2018).

¹<https://www.swisstext.org/>

²<http://commoncrawl.org/>

CNN/DailyMail This dataset is an English single-document summarization (SDS) news dataset consisting of 311,971 news articles with an average length of ~ 800 words from the CNN and DailyMail websites including abstractive summaries.

Multi-News The Multi-News dataset is an English MDS news dataset consisting of 56,216 summaries and over 250,000 sources with an average of $\sim 2,100$ words from 1,500 different sites. The summaries are linked to 2-10 human-written source documents retrieved from <https://www.newser.com/>.

auto-hMDS This is the largest dataset for multi-document summarization in German language with 2,210 summaries and 10,454 source documents, and diverse in nature. The dataset is created by selecting available summaries from Wikipedia and search for corresponding source documents on the internet. On an average, a summary is linked to 4.73 source documents.

4 Methodology

We consider the state-of-the-art BART model (Lewis et al., 2020) for the multi-document summarization (MDS) task. First, we use only pre-trained BART, and next, we fine-tune the pre-trained BART model using each of the three datasets separately and analyze the performances. The details about the BART model are described below.

Description of BART model BART (Lewis et al., 2020) generalizes the concepts of bidirectional encoders from BERT (Devlin et al., 2019) and autoregressive decoders from GPT-2 (Radford et al., 2019). The model is trained with text corrupted through an arbitrary noising function and a sequence-to-sequence model that learns to reconstruct the original text. The encoder reads the sequential input e.g. a document to summarize while the decoder generates the outputs autoregressively. Both layers are connected by cross-attention where each decoder layer focuses on specific aspects over the final state of the encoder output creating sequences, closely connected to the initial input. The bidirectional encoder architecture takes all previous and subsequent tokens into account for predicting a masked token. In text generation, BERT without any modification loses its strength of bi-directionalism and becomes directional to-

wards past words, as following words have yet to be generated. Here BART adopts the architecture of GPT-2 to predict future words only by utilizing previous words. The advantage of BART therefore is the combination of contextual embeddings from BERT and text generation from GPT-2. Transformation, as described in Lewis et al. (2020), can be implemented through token masking, token deletion, text infilling, sentence permutation, or document rotation.

Note that, in the work done by Lewis et al. (2020), authors apply the BART model only on single-document summarization (SDS) task, not on the multi-document variant of the summarization task. Therefore, to adapt the BART model for the MDS task, we follow the approach prescribed by Lebanoff et al. (2018), where authors reuse the existing SDS model for MDS by merging multiple-input to single-input. On the other hand, the issue of redundant and overlapping information is one major point to be taken care of for any summarization task, especially for MDS tasks. For that purpose, we rely on the n-gram blocking approach following the work done by Paulus et al. (2017).

5 Experimental Results and Error Analysis

For our experiments we make use of the pre-trained BART model³ and fine-tune the model on the three datasets and compare the performance with competitive baselines.

Method	R-1	R-2	R-L
LEAD-3 (Liu and Lapata)	40.42	17.62	36.67
BERTSUMABS (Liu and Lapata)	41.72	19.39	38.76
BERTSUMEXTABS (Liu and Lapata)	42.13	19.60	39.18
BART pre-trained	25.98	11.26	17.50
BART fine-tuned	42.21	19.10	35.38

Table 1: Performance of the BART pre-trained and fine-tuned models along with most competitive baselines (Liu and Lapata, 2019) on CNN/DM dataset.

5.1 Comparative Evaluation

We split each dataset into training (80%), validation (10%) and test (10%) set. In our experimental setup, we use the beam size of 4, n-gram size of 3 and use the Adam optimizer (Kingma and Ba, 2014)⁴. To evaluate the the model generated summaries, we use the variants (**R-1**, **R-2**, **R-L**) of ROUGE metric

³<https://github.com/pytorch/fairseq/tree/master/examples/bart>

⁴Default settings $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and a learning rate of $3e - 05$

(Lin, 2004) as required for the comparison with the baseline models.

Method	R-1	R-2	R-L
HI-MAP (Fabbri et al.)	40.08	14.90	19.70
BART DYNE-1 (Hokamp et al.)	43.90	15.80	22.20
BART DYNE-5 (Hokamp et al.)	43.20	13.60	20.40
BART pre-trained	30.67	10.05	16.99
BART fine-tuned	40.58	15.50	21.73

Table 2: Performance of the pre-trained and fine-tuned BART model along with baseline models on Multi-News Dataset.

Method	100 words		200 words	
	R-1	R-2	R-1	R-2
RANDOM (Zopf)	18.57	1.85	25.53	3.25
LEAD (Zopf)	12.29	2.61	10.56	2.28
TOP-5 SENTENCES	21.71	4.28	19.61	3.87
LEXRANK	29.76	6.58	23.81	5.61
BART pre-trained	28.48	8.79	20.84	6.02
BART fine-tuned	38.43	12.93	30.24	9.09

Table 3: Performance of the BART pre-trained and fine-tuned model along with baseline models on auto-hMDS dataset.

Summary (gold)
A South Carolina man says he spent 66 days alone at sea before being rescued . Other sole survivor stories include a Japanese man washed away by a tsunami . An El Salvador man says he drifted from Mexico to Marshall Islands over a year .
Summary (generated) R-1 = 12.50 R-2 = 2.53 R-L = 8.75
Sailors can't bank on technology or the proximity of a nearby city, town, or boat. In order to survive, they can rely on ingenuity, resourcefulness and luck. Jose Salvador Alvarenga says his journey began in Paredon Viejo, a port on Mexico's Pacific coast, in late 2012. He says he drank rainwater and when there wasn't any available, his own urine. Louis Jordan says he used laundry to trap and scoop up fish, rigged a makeshift mast, and sail.

Table 4: Example of missing facts within the CNN/DM dataset.

Table 1 shows the performance of the pre-trained and fine-tuned BART model on the CNN/DM dataset, along with the performance of the baseline models. We see, that the fine-tuned BART model produces comparable performance with the baselines. On the other hand, Table 2 shows the results of the pre-trained and fine-tuned BART model on the Multi-News dataset. We observe the fine-tuned model outperform the HI-MAP (Fabbri et al., 2019) model, whereas it produces comparable performance with BART-DYNE (Hokamp et al., 2020). Note that, the fine-tuned BART model considers all source documents for the MDS task whereas the model by Hokamp et al. (2020) only takes one (DYNE-1) or five source documents (DYNE-5) into account, which otherwise simplifies the task. Table 3 shows the results on the German auto-hMDS dataset of pre-trained and fine-tuned BART models in comparison to baselines proposed by Zopf (2018). We prepare two baseline models as well. The first one is trivial by extracting 'Top-5 Sentences' based on the frequency of

occurring words and the second one by following the LexRank (Erkan and Radev, 2004) approach. We see that the fine-tuned BART model outperforms all the baseline models by a significant margin, producing a state-of-the-art performance for the German MDS task⁵.

5.2 Error Analysis

Even though the BART model produces satisfactory performance for multi-document summarization for both languages, there is still scope for improvement. Hence, we investigate cases further, where even the fine-tuned BART model goes wrong. We perform this analysis for both English and German languages. To start with, we observe some interesting cases for which the model does not generate the desired gold summary due to the fact that some information in the gold summary is actually not present in any of the source documents. Table 4 represents one such interesting error case obtained from the CNN/DM dataset.

Table 5 shows one example from another frequently occurring genre of erroneous cases, for the Multi-News dataset (at top) where the model generated summary is very meaningful and comprehensive but makes up new facts such as the death of Bob Dylan (color-coded in orange). We perform a manual survey on the randomly selected model-generated summaries and observe at least 4 out of 50 summaries which include made-up facts in an otherwise coherent summary. This very pattern can also be seen while experimenting with the German auto-hMDS dataset (Table 5, at the bottom), where the place and date of birth are made-up facts. This is misleading as wrong facts are embedded in a reasonable and correct context, making them especially hard to spot.

<p>Summary (model generated) R-1 = 67.59, R-2 = 29.91, R-L = 31.41</p> <p>[...] The former James Bond star, 65, who was trained as a commercial artist and worked as an illustrator, just auctioned off one of his paintings for \$1.4 million, depicting the singer, who died in 2013. Other auction highlights included a Pierce Brosnam original painting, which sold for</p>
<p>Summary (model generated) R-1 = 55.88, R-2 = 11.94, R-L = 30.88</p> <p>Andrew Johnson (* 29. Dezember 1808 in Raleigh (North Carolina, USA; † 15. April 1865 in Greeneville, Tennessee) war der dritte Vizepraesident der Vereinigten Staaten, der durch den Tod seines Vorgaengers ins Amt kam und der erste nach einem Attentat. Als Hauptaufgabe seiner Praesidentschaft galt die sogenannte Reconstruction, der Wiederaufbau [...]</p>

Table 5: Examples of summaries showing wrong facts while experimenting with the Multi-News (Top) and auto-hMDS (Bottom) datasets.

⁵Note that, we do not report the R-L score in Table 3, as R-L scores are not reported for the baseline models used for comparison in the previous works.

<p>Summary (model generated) R-1 = 75.81, R-2 = 31.14, R-L = 32.53</p> <p>Die Westminster Abbey ist die Kroenungskirche der bristischen Monarchen seit Wilhelm dem Eroberer im 11. Jahrhundert. Erbaut wurde die Westminster Abbey zwischen 1045 und 1065 auf dem Kloster Kloster der Themse an den damals noch sumpfigen Ufern der themse errichtet. Bis zum Jahr 1529 diente der Palast den britischen Koenigen als Residenz. Heute ist der neugotische Palast vor allem als Houses of Parliament bekannt.</p>

Table 6: Example of summary showing wrong contextualization and topic extraction while experimenting with auto-hMDS dataset.

Another genre of erroneous summaries, which we detect while experimenting with the German auto-hMDS dataset, comes from lacking clear contextualization and topic delimitation. Table 6 presents one such example, where the model should summarize information about the ‘Palace of Westminster’, but as the source document includes references to related buildings, the model lost attention and mixed up information about the ‘Palace of Westminster’ and ‘Westminster Abbey’ (in orange) in one summary.

5.3 Analysis of Extractiveness of Summaries

After analyzing and pointing out the erroneous cases, we further investigate the nature of model-generated summaries along with the gold summaries of each dataset, in terms of extractiveness. Even though according to one of the recent studies (Lewis et al., 2020), the BART model output is “highly abstractive, with few phrases copied from the input”, our findings are contrary with summaries mainly built from extractive fragments or even whole paragraphs.

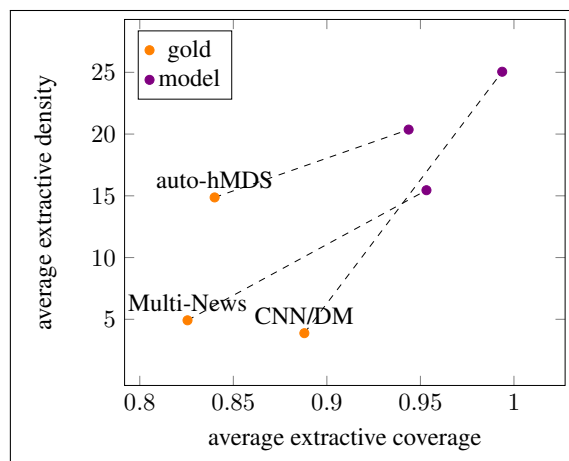


Figure 1: Comparison of extractiveness of gold summaries and model-generated summaries.

To investigate quantitatively, we measure the extractiveness by using the method of extractive coverage and extractive density, introduced by Grusky

et al. (2018)⁶.

From Figure 1, we can see that the model-generated summaries from fine-tuned BART are much more extractive than their gold counterparts with an average extractive coverage over 94%. While the gold summaries are already much more extractive, BART generated summaries increase extractiveness further. The figure also discloses the difference between the German auto-hMDS dataset and the English datasets. The average extractive density of the gold summaries from the German auto-hMDS shows that the summaries are mainly built from long extractive fragments, much more than the English gold standard summaries.

6 Conclusion

In this paper, we investigated the performance of one of the most recent pre-trained language models namely BART, for multi-document summarization tasks in English and German language. For the first time ever, we attempted fine-tuning BART for German language multi-document summarization and achieved state-of-the-art performance. We further analyzed the erroneous cases for both English and German language and attempted to find a set of patterns where BART went wrong. The insights obtained via this error analysis give rise to devise more sophisticated methods for the task of multi-document summarization addressing these errors, of which the most severe is the hallucination of facts.

Our code and data repository is available publicly⁷.

References

Regina Barzilay and Kathleen R McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Darina Benikova, Margot Mieskes, Christian M. Meyer, and Iryna Gurevych. 2016. Bridging the gap between extractive and abstractive summaries: Creation and evaluation of coherent extracts from heterogeneous sources. In *Proceedings of COLING*

⁶Extractive density measures how well a sequence of a summary is made of extractions from the source, while extractive coverage measures, how much the summary is a derivative of the source taking into account individual words. Note that, many individual words could result in a high coverage, but a high density can only be achieved with long extractive fragments.

⁷<https://github.com/uhh-1t/multi-summ-german>

2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 1039–1050, Osaka, Japan.

Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca J Passonneau. 2015. Abstractive multi-document summarization via phrase selection and merging. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1587–1597, Beijing, China.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany.

John M. Conroy, Judith D. Schlesinger, and Dianne P. O’Leary. 2006. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 152–159, Sydney, Australia.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, USA.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Alexander Richard Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium.

Jade Goldstein, Vibhu O Mittal, Jaime G Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *NAACL-ANLP 2000 Workshop: Automatic Summarization*, pages 40–48, Seattle, Washington, USA.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics:*

- Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana, USA.
- Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, Boulder, Colorado, USA.
- Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 1693–1701, Montreal, Quebec, Canada.
- Chris Hokamp, Demian Gholipour Ghalandari, Nghia The Pham, and John Glover. 2020. Dyne: Dynamic ensemble decoding for multi-document summarization. *arXiv preprint arXiv:2006.08748*.
- Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. 2020. What have we achieved on text summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469, Online.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4131–4141, Brussels, Belgium.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online.
- Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging graph to improve abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6232–6243, Online.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain.
- Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Łukasz Kaiser, and Noam Shazeer. 2018. Generating Wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations (ICLR)*, Vancouver, British Columbia, Canada.
- Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3721–3731, Hong Kong, China.
- Kathleen McKeown and Dragomir R Radev. 1995. Generating summaries of multiple news articles. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–82, Seattle, Washington, USA.
- Kathleen R McKeown, Judith L Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulation: progress and prospects. In *Proceedings of the sixteenth AAAI Conference on Artificial Intelligence*, pages 453–460, Orlando, Florida, USA.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany.
- Ani Nenkova and Kathleen McKeown. 2011. *Automatic summarization*. Now Publishers Inc.
- Shantipriya Parida and Petr Motlicek. 2019. Abstract text summarization: A low resource challenge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5994–5998, Hong Kong, China.
- Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.
- Dragomir Radev and Kathleen McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500.
- Dragomir R Radev, Hongyan Jing, and Malgorzata Budzikowska. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization*, pages 21–30, Seattle, Washington, USA.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.
- Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal.
- Chao Shen and Tao Li. 2010. Multi-document summarization via the minimum dominating set. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 984–992, Beijing, China.
- Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181, Vancouver, British Columbia, Canada.
- Christopher Tauchmann and Margot Mieskes. 2020. Language agnostic automatic summarization evaluation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6656–6662, Marseille, France.
- Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, British Columbia, Canada.
- Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.
- Markus Zopf. 2018. Auto-hMDS: Automatic construction of a large heterogeneous multilingual multi-document summarization corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, pages 3228–3233, Miyazaki, Japan.

Grammatical Error Generation Based on Translated Fragments

Eetu Sjöblom and Mathias Creutz and Teemu Vahtola

Department of Digital Humanities, Faculty of Arts, University of Helsinki, Finland
{eetu.sjoblom, mathias.creutz, teemu.vahtola}@helsinki.fi

Abstract

We perform neural machine translation of sentence fragments in order to create large amounts of training data for English grammatical error correction. Our method aims at simulating mistakes made by second language learners, and produces a wider range of non-native style language in comparison to state-of-the-art synthetic data creation methods. In addition to purely grammatical errors, our approach generates other types of errors, such as lexical errors. We perform grammatical error correction experiments using neural sequence-to-sequence models, and carry out quantitative and qualitative evaluation. A model trained on data created using our proposed method is shown to outperform a baseline model on test data with a high proportion of errors.

1 Introduction

Grammatical error correction (GEC) is the task of detecting and correcting grammatical errors in texts, typically written by second language learners. Current state-of-the-art GEC approaches are based on neural machine translation (NMT) (Grundkiewicz et al., 2019). As in other natural language processing tasks, neural approaches to GEC rely on large quantities of task-specific data, that is, sentence pairs consisting of erroneous source text coupled with corrected target text. However, in-domain GEC data is scarce, and a number of solutions to the data sparsity problem have been proposed recently, often by introducing artificially created GEC data into the training process.

Some error generation approaches also depend on error-annotated authentic learner data. For example, Felice and Yuan (2014) introduce errors probabilistically with error probabilities that are estimated using a learner corpus. Rozovskaya

et al. (2014) train error detection and classification models on annotated data, focusing on verb errors. Other methods dispense with the need for annotated data, such as approaches based on inverted spell-checkers and heuristic error generation (Grundkiewicz et al., 2019; Grundkiewicz and Junczys-Dowmunt, 2019).

To alleviate the data sparsity problem, in this work we propose to use NMT to produce artificial training data, simulating real errors made by language learners. For instance, to produce English text with errors, we use NMT models to translate sentence fragments from other languages to English, and then combine the translated fragments to form our erroneous source data. Similar machine translation approaches to GEC data creation have been proposed before. For example, Rei et al. (2017) use a statistical machine translation model trained on reversed learner data, using the corrected sentences as source data and erroneous sentences as targets. Kasewa et al. (2018) extend this approach and use an NMT model to produce errors. Htut and Tetreault (2019) perform extensive experiments on several neural models, likewise trained on learner data to generate errors.

Our contribution is to split the foreign-language source sentences into shorter fragments in order to limit the context that is available to the machine translation system. The rationale for doing this is to produce text that contains artefacts from the foreign language. Since the NMT system needs to translate shorter fragments without the proper context, we expect it to produce more literal translations and to be less able to produce correct agreement between different parts of speech. Additionally, polysemy may prompt the system to suggest translations of a synonym in the foreign language, which is not a synonym in English. The creation of synthetic training data involves further steps, which are described in Section 2. Model training is explained in Section 3. In Section 4 we evaluate our

approach quantitatively against a strong baseline (Grundkiewicz et al., 2019) and make some qualitative assessments.

2 Training data

The creation of our training data involves the following steps:

1. English sentences aligned with sentences of other languages are used as data.¹ Our parallel text data are retrieved from the OpenSubtitles (Lison and Tiedemann, 2016) and Europarl (Koehn, 2005; Tiedemann, 2012) collections.²
2. The non-English sentences are split randomly into chunks of an average length of three word tokens.
3. Each sentence chunk in isolation is translated into English using OPUS-MT machine translation models from HuggingFace (Tiedemann and Thottingal, 2020). N-best lists containing up to ten alternate translations for each chunk are produced.
4. Full English sentences are created by concatenating chunks from the n-best lists. Ten different alternate full sentence translations are generated for each source sentence by combining chunks at random, proportionally to the translation scores of the chunks. Our aim is to obtain English translations that contain errors influenced by the source language. The original English sentence from the parallel corpus serves as a correct reference translation. Examples are shown in Table 1.
5. In theory, for each sentence in our data we now have ten artificially created, erroneous English sentences. However, many of the synthetic sentences do not resemble authentic human-produced erroneous sentences. We therefore discard a significant portion (60 %) of the synthesized sentences by sampling for an error distribution that is closer to the error distribution of authentic data, represented by our development sets. This leaves us with just

¹These languages, which have been chosen to represent both European and Asian languages from different language families are the following: Danish, Dutch, Finnish, French, German, Italian, Japanese, Korean, Latvian, Portuguese, Russian, Spanish, and Swedish.

²Available for download at: <https://opus.nlpl.eu/>

23 % of the words of our original set, reflecting the fact that longer sentences are more likely to be discarded.

6. The above mentioned sampling of sentences requires us to be able to compare error distributions between authentic and synthetic data. First we POS tag the sentence pairs and align them automatically using minimum string edit distance coupled with some heuristics taking into account part of speech and inflection. The alignment algorithm is similar but not identical to ERRANT (Bryant et al., 2017; Felice et al., 2016). This procedure is illustrated in Table 2. From the alignments we extract trigrams consisting of a correction operation in the context of one preceding and following token, such as *PRP ins(MD) VBP* (“insert a modal verb between a pronoun and a verb in non-third person singular form”) or *ins(DT) ins(JJ) NN* (“insert an adjective between an inserted determiner and a noun in singular”). These automatically extracted trigrams constitute our correction types. Their frequency distributions are not the same across the authentic and synthetic data. We filter the synthetic data by keeping sentence pairs that contain combinations of correction types that are highly likely to occur in authentic data and discard sentence pairs with low-probability correction types.

2.1 Final training sets

We carry out experiments using systems trained on four different training sets. We create one data set using our method that matches the word count of the Baseline comparison. In addition, we create two smaller data sets using both the Baseline method and ours on the same correct target sentences in order to control the effects of data domain.

- **Baseline:** We compare our own training scheme to a system trained on the training set created by Grundkiewicz et al. (2019). They propose an unsupervised data generation method based on confusion sets from spellcheckers. For each source sentence in the news crawl data used for training, they replace a random number of tokens with a substitute from the vocabulary item’s confusion set. In

de-src	Während / du / bewusstlos im / Krankenhaus / lagst, sind / die Ärzte mit / diesen / Testergebnissen / zu / Daniel gekommen.
en-tgt	During / you / unconscious in / Hospital / the / doctors with / the / Test results / to / Daniel came.
en-ref	While you were unconscious in the hospital, the doctors came to Daniel with these test results.
ru-src	И никогда не / переставал / думать о / тебе.
en-tgt	And never / I stopped / to think about / You.
en-ref	I never stopped thinking about you.
fr-src	Il est / vrai que toutes les / histoires ne peuvent avoir une fin heureuse, mais pour Jules / Daly, la rêveuse de Buffalo, / l'histoire ne / fait que commencer.
en-tgt	It is / true that all / stories can't have a happy ending, but for Jules / Daly, Buffalo's dreamer, / history / Just start.
en-ref	It is true not all tales have happy endings, but then for Jules Daly, the dreamer from Buffalo, the story is just beginning.
ja-src	もし君が生き残れたら / 一生懸命に働いたからだ
en-tgt	And if you survive, / Because you worked hard.
en-ref	If you live, you have worked very hard indeed.
fi-src	Koulu / on - / lähettänyt / minut useammalle / terapeutille kuin / sinulla / on ollut huonoja / treffejä.
en-tgt	School / is / sent / me to more / for a therapist / you / has been bad / Date.
en-ref	This school has sent me to more therapists than you've had bad dates.

Table 1: Sentences in other languages (*-src) are split into chunks (e.g., / *bewusstlos im I*), and each chunk in isolation is translated automatically into English. By concatenating the chunks we obtain English sentences containing errors (en-tgt), for which correction hypotheses exist in the form of the English reference translations (en-ref).

addition, they probabilistically delete and insert random tokens, as well as swap adjacent tokens in the sentence. They also introduce additional noise at the character level using similar operations. Although these operations introduce some syntactic and word order mistakes, the method does not excel at producing more complex syntactic errors, errors that require extensive reordering of the sentence, or errors that result from L1 influence.

- **Chunks:** We produce a training set using our method, which is sampled to contain the same number of words as the Baseline (4.6 billion words).
- **Chunks-small:** We produce a training set using our method such that the data set contains only unique target sentences. This smaller set contains approximately 650 million word tokens and allows for faster model training.
- **Baseline-small:** We use the Baseline data creation method on the same target sentences as in the Chunks-small set.

3 Model training

We build on the system described in Grundkiewicz et al. (2019). We choose not to make changes to the model or training parameters in order to isolate the effects of our data creation method and ensure a fair comparison. The same training setup is used for

all models, with modifications only in the training sets. Specifically, we use their “Transformer Big” architecture, with 6 self-attention layers, 16 attention heads, embeddings vectors of size 1024, and feed-forward hidden size of 4096 with ReLU activation functions. We also tie the encoder, decoder, and output embeddings.

We also adopt the training setup of Grundkiewicz et al. (2019), and train our models with the Marian toolkit (Junczys-Dowmunt et al., 2018). The models are first pretrained on the synthetic data for a maximum of 5 epochs. After pretraining, we finetune the best model checkpoint using the following corpora: FCE (Yannakoudakis et al., 2011), NUCLE (Dahlmeier et al., 2013), W&I-LOCNESS (Bryant et al., 2019; Granger, 1998), and Lang-8 (Mizumoto et al., 2012). We use the W&I-LOCNESS development set for validation during training.

We use early stopping with a patience of 10 with ERRANT $F_{0.5}$ score on the W&I+locness development set used as the early stopping criterion. The checkpoint with the highest $F_{0.5}$ score is chosen for further finetuning. We choose the ADAM optimizer, a learning rate of 0.0002 and a linear warmup for 8k updates. We use Marian’s option to dynamically fit mini-batches to GPU memory, and train our models using 4 Nvidia Volta V100 GPUs (32GB RAM). In addition, we use strong regularization, which has been found useful in GEC systems, with dropout probabilities of 0.3 between

Learner sentence:	We had enjoy time .
Correction:	We had a great time .
Alignment:	PRP VBD del(VB) ins(DT) ins(JJ) NN .
Synthetic sentence:	You be the the old donkey of the forestry
Correct reference:	You 'll be the oldest donkey in the forest .
Alignment:	PRP ins(MD) VBP del(DT) DT inf(JJS) NN del(IN) ins(IN) DT typ(NN) ins(.)

Table 2: Pairs of sentences with alignments. The upper example is an authentic sentence produced by a language learner accompanied by a correction (target hypothesis) proposed by a teacher. The alignment is a sequence describing how to modify the learner sentence into the corrected one. It reads as follows: *PRP*: keep pronoun (“we”), *VBD*: keep verb in past tense (“had”), *del(VB)*: delete verb in infinitive (“enjoy”), *ins(DT)*: insert determiner (“a”), *ins(JJ)*: insert adjective (“great”), *NN*: keep noun in singular (“time”), *.*: keep punctuation. The lower example is analogous, but the alignment is between a synthetically produced sentence and the correct English reference. This alignment sequence contains a few more correction types: *inf(JJS)*: change inflection of adjective into superlative (“oldest”), *typ(NN)*: fix typo in noun in singular (the word “forestry” is here classified as a spelling error by the algorithm).

	W&I-LOCNESS	YKI
Baseline	66.44	52.63
Chunks	65.44	53.41
Baseline-small	60.09	46.66
Chunks-small	59.89	49.73

Table 3: $F_{0.5}$ scores for the four models on the two test sets. $F_{0.5}$ is a weighted harmonic mean of precision and recall, where precision is accentuated.

layers, 0.1 for self-attention and feed-forward layers, 0.3 for entire source token embeddings, and 0.1 for target embeddings.

4 Evaluation

We report results on two different data sets. We use the W&I-LOCNESS set, which was used as official test data in the BEA19 GEC shared task (Bryant et al., 2019), as well as a subset of the English portion of learner texts derived from the Finnish National Certificates of Language Proficiency exams (YKI).³ We do not use the YKI data as a blind test set, but instead use it to qualitatively analyze differences in model predictions. Still, no part of the YKI data was used during training or development of the models.

We compare our results with those reported by Grundkiewicz et al. (2019), whose system achieved first place in the BEA19 GEC shared task. However due to limited resources we do not train an ensemble of models, but instead take a single left-to-right

³Available for research purposes from the Centre for Applied Language Studies at the University of Jyväskylä, Finland: <http://yki-korpus.jyu.fi/>

model from Grundkiewicz et al. (2019) as baseline. Their best system uses an ensemble approach with right-to-left and language model reranking and achieves a higher $F_{0.5}$ score of 69.47 on the W&I-LOCNESS test set.

The upper part of Table 3 compares our Chunks model with the baseline by Grundkiewicz et al. (2019). The Baseline model performs best on W&I-LOCNESS with a one absolute point difference compared to our model. However, our model outperforms the Baseline on YKI. These results suggest that our data creation method might be suitable when correcting noisier source sentences, as YKI generally contains more challenging language with more errors than W&I-LOCNESS.

The lower part of Table 3 demonstrates that the trends are the same for the smaller models, in which we match the data domain in training. That is, the Baseline is no longer trained on news data but on OpenSubtitles and Europarl. The results are lower overall due to the smaller data size. The Baseline outperforms our model on W&I-LOCNESS also in this setting, although by a smaller margin. However, the performance gap on YKI increases by approximately two absolute points in favor of our model, offering additional support that our method can improve performance on noisy data. To better understand differences between the models, we examine their predictions on the same source sentences, as described in the next section.

4.1 Qualitative assessment

We have taken a closer look at the corrections made by the Baseline and Chunks models on the 320

sentences in the YKI data set. The results are surprisingly similar although the models have been trained on different text corpora, into which errors have been introduced using different methods. The Chunks model suggests slightly more corrections on average than the Baseline, yielding a somewhat higher recall and lower precision.

It is interesting to see that the Chunks model performs quite well on misspelled words (*broblems*, *i'ts*, *beatyfull*) although it has not been explicitly trained to correct spelling mistakes, in contrast to the Baseline method. In the training data of the Baseline, spelling errors have been introduced by random sampling, whereas the models based on machine translated data generally do not contain any spelling mistakes, as machine translation does not generate them. Yet, it appears that the Chunks model corrects spelling errors at least as well, if not better, than the Baseline. The latter model leaves word forms, such as *wery*, *nicier* and *higing* (for *hiking*) unchanged.

When it comes to choosing the correct spelling in context, the Chunks model distinguishes between the different usages of *prize* and *price* (“*The prize for you is between 1500-1700 euros.*”), and it has in fact been trained on almost 4000 sentence pairs in which *prize* is corrected to *price* in context. The Baseline does not make this correction.

None of the models manage to correct the sentence “*I have old but wery fine cun selling.*”. Firstly, the models fail to change *cun* into *gun*. Secondly, one could have expected the Chunks model to see the connection between *selling* and *for sale*, since there are 800 training examples containing that substitution, but for some reason this particular test sentence does not trigger the desired change.

Many of the sentences in the YKI corpus are indeed hard to interpret without broader world knowledge; The Chunks model corrects the sentence “*If it isn't help then you will ask for help to polishman*” into “*If it doesn't help then you will ask the Polishman for help.*”. However, the correct person to ask for help here would be the police man. In another sentence, “*I begin hobbies about 12 yers.*”, the model would somehow need to understand that the person picked up hobbies at the age of twelve rather than twelve years ago.

Additionally, we have examined the W&I-LOCNESS development set, although it has been used as a stopping criterion in the training, which may bias the results slightly. The $F_{0.5}$ scores on


the dev set are the same for both the Baseline and the Chunks model (52.6 %). This is considerably lower than for the final test set (65.4 - 66.4 %), suggesting that the test set is less challenging than the dev set. Compared to the YKI data, even the W&I-LOCNESS dev set seems cleaner and appears to contain fewer mistakes. It is hard to see significant differences in performance between the models. For 65 % of the sentences, the Baseline and the Chunks model produce exactly the same corrections. The corresponding figure for the YKI set is 57 %.

5 Discussion and conclusion

We have shown that our model rivals a competitive baseline, a left-to-right model by Grundkiewicz et al. (2019), which was one component of an ensemble model that performed best in the BEA19 GEC shared task. We did not yet train our own ensemble model, but we expect to see similar improvements in performance in future experiments.

Our results show that two models can perform on par, although they have been pretrained on different training corpora and using different error simulation techniques. In addition, the Chunks model outperforms the Baseline in noisy conditions. In the future, we would like to analyze further techniques for modeling challenging types of errors, which originate from structures that differ between the target language and the native languages of the language learners.

6 Acknowledgments

 This study has been supported by the FoTran project, funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement № 771113). We wish to acknowledge CSC – IT Center for Science, Finland, for generous computational resources.

References

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. <https://doi.org/10.18653/v1/W19-4406> The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction.
- Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. <https://www.aclweb.org/anthology/W13-1703> Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.
- Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. Automatic extraction of learner errors in esl sentences using linguistically enhanced alignments.
- Mariano Felice and Zheng Yuan. 2014. <https://doi.org/10.3115/v1/E14-3013> Generating artificial errors for grammatical error correction. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–126, Gothenburg, Sweden. Association for Computational Linguistics.
- Sylviane Granger. 1998. The computer learner corpus: a versatile new source of data for sla research. In *Learner English on Computer*, pages 3–18. Addison Wesley Longman.
- Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2019. <https://doi.org/10.18653/v1/D19-5546> Minimally-augmented grammatical error correction. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 357–363, Hong Kong, China. Association for Computational Linguistics.
- Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. <https://doi.org/10.18653/v1/W19-4427> Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.
- Phu Mon Htut and Joel Tetreault. 2019. <https://doi.org/10.18653/v1/W19-4449> The unbearable weight of generating artificial errors for grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 478–483, Florence, Italy. Association for Computational Linguistics.
- Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. <https://doi.org/10.18653/v1/W18-2716> Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135, Melbourne, Australia. Association for Computational Linguistics.
- Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. <https://doi.org/10.18653/v1/D18-1541> Wronging a right: Generating better errors to improve grammatical error detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4977–4983, Brussels, Belgium. Association for Computational Linguistics.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proc. MT Summit 2005*.
- Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.
- Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. <https://www.aclweb.org/anthology/C12-2084> The effect of learner corpus size in grammatical error correction of ESL writings. In *Proceedings of COLING 2012: Posters*, pages 863–872, Mumbai, India. The COLING 2012 Organizing Committee.
- Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. <https://doi.org/10.18653/v1/W17-5032> Artificial error generation with machine translation and syntactic patterns. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 287–292, Copenhagen, Denmark. Association for Computational Linguistics.
- Alla Rozovskaya, Dan Roth, and Vivek Srikumar. 2014. <https://doi.org/10.3115/v1/E14-1038> Correcting grammatical verb errors. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 358–367, Gothenburg, Sweden. Association for Computational Linguistics.
- Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).
- Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. <https://www.aclweb.org/anthology/P11-1019> A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Creating Data in Icelandic for Text Normalization

Helga Svala Sigurðardóttir
Reykjavik University
helgas@ru.is

Anna Björk Nikulásdóttir
Grammatek
anna@grammatek.com

Jón Guðnason
Reykjavik University
jg@ru.is

Abstract

We introduce Regína, a rule-based system that can automatically normalize data for a text-to-speech (TTS) system. Normalized data do not generally exist so we created good enough data for more advanced methods in text normalization (TN). We manually annotated the first normalized corpus in Icelandic, 40,000 sentences, and developed Regína, a TN-system based on regular expressions. The new system gets 89.82% accuracy compared to the manually annotated corpus on non-standard words and showed a significant improvement in accuracy when compared to an older normalization system for Icelandic. The normalized corpus and Regína will be released as open source.

1 Introduction

Text normalization is an integral part of a TTS system. Unrestricted input texts can contain so-called non-standard words (NSWs), which are impossible for a computer to read without being formatted into regular strings of alphabetical letters and punctuation marks. These NSWs are divided into semiotic classes and include abbreviations, numbers, and special characters.

The degree of importance of text normalization in TTS is not obvious even though its utility is known. Most words do not need to be normalized, and therefore normalized datasets and their unnormalized counterparts are almost identical. However, without expanding NSWs, a TTS system skips those words, making the text inaccurate and incomplete.

To clarify, let us look at an example of a sentence before and after normalization.

Hæsti tindur Esjunnar er 914 m.
(Esjan's highest peak is 914m.)

↓

Hæsti tindur Esjunnar er níu
hundruð og fjórtán metrar.
(Esjan's highest peak is nine
hundred and fourteen meters.)

Text normalization systems are customarily rule-based but are moving in the direction of neural networks (NNs). Models made with NNs require less human effort (Graves and Jaitly, 2014) but need a vast amount of correctly annotated data to learn from, and these do not naturally exist for text normalization. People can generally read NSWs without requiring an explanation, so there is no motivation to create data with normalized text, such as in translation. To acquire data in Icelandic for the training of more sophisticated systems, we start by making a system that can make data good enough for further training. We compare the results of this system with manually annotated data to better assess the quality.

1.1 Background

In 1996, Sproat (Sproat, 1996) published work for a unifying model for most text normalization problems, built with Weighted Finite-State Transducers (WFSTs). The transducers were constructed using a lexical toolkit that allows descriptions of lexicons, morphological rules, numeral-expansion rules, and phonological rules. In 2001, Sproat (Sproat et al., 2001) expanded on this work and described challenges that heavily inflected languages like Russian (and Icelandic) face. This work was the first that treated the problem as essentially a language modelling problem.

Up until recently, the primary approach to the text normalization problem was with WFSTs. In 2015, Ebden et al. (Ebden and Sproat, 2015) released a paper where they described the Kestrel text normalization system, a component of the Google TTS system. It differed from previous systems by separating the tokenization and classifica-

tion (determining whether a word should be normalized and, if so, which semiotic class it belongs to) from the verbalization step. Kestrel recognizes a large set of semiotic classes: various categories of numbers, times, telephone numbers and electronic addresses.

Work on Icelandic spoken language technologies is defined within the Language Technology Programme for Icelandic (2019-2023) (Nikulásdóttir et al., 2020). Previous work on language resources for Automatic Speech Recognition (ASR) and TTS include acoustic data gathering (Guðnason et al., 2012; Steingrímsson et al., 2017; Mollberg et al., 2020) and text corpus building for Icelandic (Steingrímsson et al., 2018). Spoken language technologies for Icelandic commenced with building ASR systems (Helgadóttir et al., 2017) with resource work on TTS aimed at a pronunciation lexicon (Nikulásdóttir et al., 2018) and acoustic data recordings (Sigurgeirsson et al., 2020) following.

The only research that has been done on text normalization in Icelandic was done in 2019, (Nikulásdóttir and Guðnason, 2019) focusing exclusively on numbers. The system built follows the open-source version of Kestrel, Sparrowhawk¹ (Ebden and Sproat, 2015), and contains a set of grammar rules written in Thrax. Numbers are handled with a classification grammar, which classifies input containing digits into several semiotic classes, and a verbalization grammar, which inflates the numbers. The verbalization grammar labels possible verbalizations with part-of-speech tags and a language model is then used to choose the most probable word form where verbalization is ambiguous.

In the last few years, people have been experimenting with deep learning (neural networks) for text normalization (Pusateri et al., 2017; Pramanik and Hussain, 2019; Zhang et al., 2019). This works well for many tasks, but the task of text normalization is fragile. Neural networks are prone to so-called unrecoverable errors; they do not only expand the words incorrectly, but the result is misleading. For instance, a navigation system could send the user to another side of town because it incorrectly expanded the postal code. Some experiments have been performed with hybrid systems, using a neural model and then applying a grammar system, such as Kestrel. The grammar system

implements an overgenerating grammar, which includes the correct verbalization, and can be used to guide the system (Sproat and Jaitly, 2017; Zhang et al., 2019, 2020).

In 2016, Sproat et al. (Sproat and Jaitly, 2016) released a challenge: given a large corpus of written text aligned to its normalized spoken form, train an RNN to learn the correct normalization function. The authors presented a dataset of general text with generated normalizations using an existing text normalization component of a TTS system (Kestrel).

2 Data

The data used are 40,000 sentences (741,909 words) from the 2017 version of the Icelandic Gigaword Corpus (IGC). We use sentences that include many NSWs, such as numbers, abbreviations, and symbols. They are from all sources in the IGC. 534 of the sentences deal with sports results and were handled separately. The sentences were manually annotated and make up the first manually curated normalization corpus for Icelandic. For a small experiment on inter annotator agreement, three people from Reykjavík University normalized 30 sentences with 205 NSWs, using the guidelines in Appendix B. The annotators expanded words without regard to a semiotic class. The inter-annotator agreement for NSWs was $\kappa = 0.85$.

3 Methodology

Icelandic is an inflected language, where each word can have various forms of words depending on the context. For example, the number 2 (*two*) can be expanded as *tveir*, *tvo*, *tveimur*, *tveggja*, *tvær*, or *tvö*, depending on the next word's case. The ordinal number 2. (*second*) can then be *annar*, *annan*, *öðrum*, *annars*, *önnur*, *aðra*, *annarri*, *annarrar*, *annað*, *öðru*, *annars*, *aðrir*, *annarra*, or *aðrar*. Only the first four numbers (one, two, three, and four) have this inflected nature.

The most significant ambiguity in the data was whether to write hyphens and dashes as *til* (to) or silence when it was used to describe sports results. In Icelandic, a sentence like *Leiknum lauk með 2-1 sigri* (The game ended with a 2-1 victory), is read as *Leiknum lauk með tvö (2) eitt (1) sigri* and the hyphen is silent. In a TTS system, the idea is that the user can either mark the topic herself or run the text through data-driven topic classification.

¹<https://github.com/google/sparrowhawk>

The system built in this research uses regular expressions and grammar rules to determine how a word should be expanded. It has been given the name Regína. The first step of Regína is to run rules for expansions of abbreviations, measurements, money, weblinks, and roman numerals through the unnormalized text. The rules for measurements take prepositions into account. For example, this could help when the base version of *km* is *kílómetrar*. If we say *til 2 km*, Regína uses the preposition *til* to expand the word to the genitive case, *kílómetra*. The next step is to run this expanded text through a part-of-speech (POS) tagger. (Steingrímsson et al., 2019) Instead of reading *km* as an abbreviation (and giving it a tag as such), the tagger now recognizes the word *kílómetra* and knows from context it is in genitive case. Now Regína is preserving part-of-speech tags for each word. Next, the semiotic class of remaining NSWs is determined. Rules for numbers are applied to cardinal and ordinal numbers, decimals and fractions. In this step, the words tagged as numbers consider the next word’s tag. The numbers that are not followed by an adjective or a noun are assigned a default case. The final step of the system is to run the text through rules for other semiotic classes: time, sports results, digits, letters, dates, and symbols. For comparison, the normalized text was re-aligned with the manually annotated text, with each sentence and word indexed to keep the structure clear. In Appendix A, the pipeline for Regína is shown.

4 Results

The dataset with general news had 729,763 words, of which 701,088 did not need normalization. The baseline of the system without any work was thus 96.08%. The remaining 28,675 words were split into cardinal, ordinal, and decimal numbers, digits, fractions, letter sequences, abbreviations, weblinks, measurements, clock times, dates, and symbols. The accuracy and size of each class are shown in Table 1.

Sports

The only specific domain looked at were sports because of the ambiguity regarding hyphens. The portion regarding sports was 12,106 words, 1.7% of the dataset. The ratio of NSWs in need of normalization is relatively high in sports, 14.66%. We looked at the same semiotic classes, with an addition of a special one for sports results.

SEMIOTIC CLASS	ACCURACY [%]	# examples
ALL	99.51	729,673
PLAIN	99.94	626,541
CARDINAL	86.87	8,456
ORDINAL	87.24	1,653
DIGIT	51.45	241
DECIMAL	74.36	197
FRACTION	33.33	39
LETTERS	96.05	3,576
ABBREVIATIONS	80.72	1,675
ROMAN NUMERALS	33.66	104
MONEY	46.89	352
WLINK	99.14	348
MEASURE	61.96	1,559
TIME	80.36	713
DATE	97.75	7,937
SYMB	88.36	1,735
PUNCT	99.93	74,547

Table 1: Results for general news

SEMIOTIC CLASS	ACCURACY [%]	# examples
ALL	98.45	12,106
PLAIN	99.98	8,923
CARDINAL	96.84	538
ORDINAL	91.89	74
DIGIT	0.0	1
DECIMAL	0.67	3
FRACTION	0.0	1
LETTERS	99.06	106
ABBREVIATIONS	75.0	20
WLINK	100.0	1
MEASURE	60.0	5
TIME	100.0	2
DATE	88.4	43
SYMB	90.91	88
SPORT	84.55	893
PUNCT	1.0	1,408

Table 2: Results for sports news

Error division

We considered error division for the classes and listed them in Table 4. All classes are handled alike in the two domains except for the symbol class (where a dash is generally a *til (to)* but silent in the sport domain), and the SPORT class is unique to sports news. The errors are divided up to:

- *CLASS* – incorrect normalization due to misclassification of the token
- *FORM* – incorrect grammatical form of the normalization but otherwise correct
- *NON-ERRORS* – errors due to errors in the manual data, misalignment of whitespaces, or instances where both expansions are correct but different (e.g. þúsund and eitt þúsund (thousand and one thousand)).

SEMIOTIC CLASS	ORIGINAL	MANUAL	MACHINE (evt. classification)	ERROR
CARDINAL	4	fjórum	fjögur	FORM
ORDINAL	2.	öðru	annað	FORM
DECIMAL	2.4	tveir komma fjórir	tveir punktur fjórir (DIGIT)	CLASS
DECIMAL	12,883	tólf þúsund átta hundruð áttatíu og þrír	einn fimm komma átta átta þrír (DIGIT)	CLASS
DATE	4/4	fjórði apríl	fjórír fjórðu (FRACTION)	CLASS
FRACTION	1/8	einum áttunda	einn áttundu	FORM
PLAIN	ALLIR	ALLIR	A L L I R (LETTERS)	CLASS
ABBREVIATION	-100 kg	undir hundrað kíló	mínus hundrað kíló (wrong word)	OTHER
CARDINAL	70s	seventies (English)	sjötíu sekúndur	OTHER
MEASURE	3 cm	þriggja sentimetra	þrír sentimetrar	FORM
TIME	1:22	eitt tuttugu og tvö	ein tuttugu og tvær	FORM
DATE	1. nóv 2012	fyrsta nóvember tvö þúsund og tólf	fyrsti nóvember tvö þúsund og tólf	FORM
SPORT	24/7	tuttugu og fjóra <sil> sjö	tuttugu og fjögur <sil> sjö	FORM
SYMB (general)	-	-	til	OTHER
SYMB (sport)	-	til	-	OTHER
PUNCT	/	/	skástrik (SYMB)	CLASS

Table 3: Incorrect results from Regína

- *NO ACTION* – the token was not expanded
- *INSUFFICIENT* – the token was only partially expanded
- *OTHER* – the token was normalized incorrectly, not due to class or grammatical form. Examples include dates written in English, incorrectly expanded dashes, and reverse order of money, such as \$5 incorrectly being expanded to *dollarar fimm (dollars five)*.

Comparison with an existing system

To compare Regína with the old Thrax normalizer, Textahaukur (Nikulásdóttir and Guðnason, 2019), 400 sentences from the whole dataset were normalized with both systems. 147 of those contained NSWs and were observed for more meaningful results. Regína had an accuracy score of 83.67%, with 20 sentences containing 22 words that did not match the manual annotation. Textahaukur had an accuracy score of 61.22%, with 55 sentences containing 106 incorrectly normalized words.

4.1 Discussion

Normalization systems are either rule-based, made with neural models or a hybrid of those two. The drawback of a rule-based system is that it is less generalizable and requires more maintenance. The main advantage is that it never makes *unrecoverable errors*. The worst errors Regína makes is not expanding a non-standard word, which happens when it does not find an appropriate semiotic class. It can also happen that it assigns the wrong class to it – making the expansion comprehensible but awkward.

As mentioned, the main problem with an inflected language like Icelandic is that each word has several forms. A part-of-speech tagger helps determine the expansion of the preceding number, but if the word following a number is not a noun or an adjective, it is given a default form. For cardinal numbers, that is the neutral, nominative, singular version, which works well with sports results, years, timings, addresses, et cetera. For decimals, it is the masculine, nominative, singular version. For ordinal numbers, it is the masculine, dative, singular form. This covers most cases, especially dates.

These default cases, plus the next word’s tag, covered a vast majority of examples in the data. The incorrect examples from these semiotic classes, as seen in 4, are mostly from the target word neither having a tag for reference nor being in the default form. Abbreviations, measurements, and fractions have the same problem, i.e., the default class is not correct. The system also marks dates written as 6/6 as fractions and expands them to *sex sjöttu (six sixths)* instead of *sjötti júní (the sixth of June)*.

The system is built with an intention of a spell-correcting layer before the normalization. In Icelandic, the rule is to write thousands separators with a dot and decimal separators with a comma, opposite to English. Regína sends numbers that do not conform to Icelandic rules to the digit class and writes them out, digit by digit, sometimes going against the author’s intention.

The time class only has rules for the 24-hour clock format, so when it read results from time-keeping, it did not expand the numbers correctly. The symbol class mostly suffers from the strict

SEMIOTIC CLASS	# ERRORS	CLASS	FORM	NON-ERRORS	NO ACTION	INSUFFICIENT	OTHER
PLAIN	384	280	0	103	0	0	1
CARDINAL	882	17	820	23	8	13	1
ORDINAL	223	6	212	0	0	5	0
DIGIT	118	110	0	4	0	4	0
DECIMAL	51	27	23	0	1	0	0
FRACTION	27	3	21	0	1	0	2
LETTERS	142	9	0	11	120	2	0
ABBREVIATIONS	328	51	83	1	184	8	1
ROMAN NUMBERS	69	0	47	0	22	0	2
MONEY	188	1	84	3	5	64	31
WLINK	4	2	0	0	1	0	1
MEASURE	595	6	524	2	0	60	3
TIME	140	4	3	1	0	132	0
DATE	182	16	81	663	4	8	11
SPORT	140	46	58	34	2	0	0
SYMB (general)	202	0	1	1	189	0	11
SYMB (sport)	8	0	0	0	0	0	8
PUNCT	53	50	0	3	0	0	0

Table 4: Error division

translation of / to *skástrik* (slash) and *-/* to *til* in general text, silence in sports. Regína tried to catch all non-standard words, sometimes outside its scope. Parts of sentences in Icelandic text are sometimes written with spelling errors, in English, or as with the separators, with rules that do not apply to Icelandic. Both ends have rigid rules about weblinks and sports results, and the results are almost 100% accurate. The only incorrect examples are misclassified – like 24/7 (twenty-four-seven) is classified as a sports result.

Finally, the slight inaccuracy of the plain class, which should remain unchanged, resulted mainly from words being misclassified to the LETTERS class (NATO → N A T O) and mistakes in the manual data.

4.1.1 Comparison between systems

The errors made by Regína and Textahaukur were examined. Regína had some abbreviations that were not expanded because of possible ambiguity. Otherwise, a majority of the errors was the wrong case of an expanded number.

These were also the most common errors for Textahaukur. More serious errors were a strong tendency to change cases in the middle of a token. For example, the number 110 was normalized in the feminine for the first part (*hundraðasta og*) and then masculine (*tíundi*). Textahaukur deleted tokens when they were followed by a token it could not handle (5,5°C became °) or skipped handling a whole sentence. In some cases, Textahaukur did not have any rules implemented. These were cases of weblinks and sports, which Regína handles almost perfectly with rigid rules on both ends.

Regína and Textahaukur both had cases where they expanded correctly, but the manual normalization was incorrect, showing that even when a computer knows less than a person, it is more consistent.

4.2 Conclusions and future work

Regína works well and does not return misleading results. The manually annotated data inevitably became a development dataset, since it was always visible for the developer of Regína. However, this is exclusively a problem for comparing the system with the corpus. Regína will be used to normalize text for TTS synthesis. Although the exact expansion might differ from person to person, that does not indicate an incorrect normalization.

In the future, we want to do more thorough experiments on inter-annotator agreement. For the 205 words, the annotators mostly disagreed on words that can be expanded in multiple ways. Regína will be used to normalize more data for further development in text normalization, using neural models. For the TTS application, we will create a test set-up for extrinsic evaluation given the new dataset.

Acknowledgements

This work is supported by the Language Technology Programme for Icelandic 2019-2023, funded by the Icelandic government. We want to thank Haukur Páll Jónsson and David Erik Mollberg for annotation and conversations about data structure, and Ari Páll Kristinsson, for advice on grammar.

References

- Peter Ebdem and Richard Sproat. 2015. The Kestrel TTS text normalization system. *Natural Language Engineering*, 21(3):333.
- Alex Graves and Navdeep Jaitly. 2014. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772. PMLR.
- Jón Guðnason, Oddur Kjartansson, Jökull Jóhannsson, Elín Carstensdóttir, Hannes Högni Vilhjálmsson, Hrafn Loftsson, Sigrún Helgadóttir, Kristín M Jóhannsdóttir, and Eiríkur Rögnvaldsson. 2012. *Almannaromur*: An open icelandic speech corpus. In *Spoken Language Technologies for Under-Resourced Languages*.
- Inga Rún Helgadóttir, Róbert Kjaran, Anna Björk Nikulásdóttir, and Jón Guðnason. 2017. Building an asr corpus using althingi’s parliamentary speeches. In *INTERSPEECH*, pages 2163–2167.
- David Erik Mollberg, Ólafur Helgi Jónsson, Sunneva Þorsteinsdóttir, Steinþór Steingrímsson, Eydís Huld Magnúsdóttir, and Jon Gudnason. 2020. Samrómur: Crowd-sourcing data collection for icelandic speech recognition. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3463–3467.
- Anna Björk Nikulásdóttir and Jón Guðnason. 2019. Bootstrapping a Text Normalization System for an Inflected Language. Numbers as a Test Case. In *INTERSPEECH*, pages 4455–4459.
- Anna Björk Nikulásdóttir, Jón Guðnason, Anton Karl Ingason, Hrafn Loftsson, Eiríkur Rögnvaldsson, Einar Freyr Sigurðsson, and Steinþór Steingrímsson. 2020. Language technology programme for icelandic 2019-2023. *arXiv preprint arXiv:2003.09244*.
- Anna Björk Nikulásdóttir, Jón Guðnason, and Eiríkur Rögnvaldsson. 2018. An icelandic pronunciation dictionary for tts. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 339–345. IEEE.
- Subhojeet Pramanik and Aman Hussain. 2019. Text normalization using memory augmented neural networks. *Speech Communication*, 109:15–23.
- Ernest Pusateri, Bharat Ram Ambati, Elizabeth Brooks, Ondrej Platek, Donald McAllaster, and Venki Nagesha. 2017. A Mostly Data-Driven Approach to Inverse Text Normalization. In *INTERSPEECH*, pages 2784–2788. Stockholm.
- Atli Sigurgeirsson, Gunnar Örnólfsson, and Jón Guðnason. 2020. Manual speech synthesis data acquisition-from script design to recording speech. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 316–320.
- Richard Sproat. 1996. Multilingual Text Analysis for Text-to-Speech Synthesis. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP’96*, volume 3, pages 1365–1368. IEEE.
- Richard Sproat, Alan W Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer speech & language*, 15(3):287–333.
- Richard Sproat and Navdeep Jaitly. 2016. RNN Approaches to Text Normalization: A Challenge. *arXiv preprint arXiv:1611.00068*.
- Richard Sproat and Navdeep Jaitly. 2017. An RNN Model of Text Normalization. In *INTERSPEECH*, pages 754–758. Stockholm.
- Steinþór Steingrímsson, Jón Guðnason, Sigrún Helgadóttir, and Eiríkur Rögnvaldsson. 2017. Málrómur: A manually verified corpus of recorded icelandic speech. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 237–240.
- Steinþór Steingrímsson, Sigrún Helgadóttir, Eiríkur Rögnvaldsson, Starkaður Barkarson, and Jón Guðnason. 2018. Risamálheild: A very large icelandic text corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Steinþór Steingrímsson, Örvar Káráson, and Hrafn Loftsson. 2019. Augmenting a bilstm tagger with a morphological lexicon and a lexical category identification step. *arXiv preprint arXiv:1907.09038*.
- Hao Zhang, Richard Sproat, Axel H Ng, Felix Stahlberg, Xiaochang Peng, Kyle Gorman, and Brian Roark. 2019. Neural Models of Text Normalization for Speech Applications. *Computational Linguistics*, 45(2):293–337.
- Junhui Zhang, Junjie Pan, Xiang Yin, Chen Li, Shichao Liu, Yang Zhang, Yuxuan Wang, and Zhenjun Ma. 2020. A hybrid text normalization system using multi-head self-attention for mandarin. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6694–6698. IEEE.

A Regína Pipeline

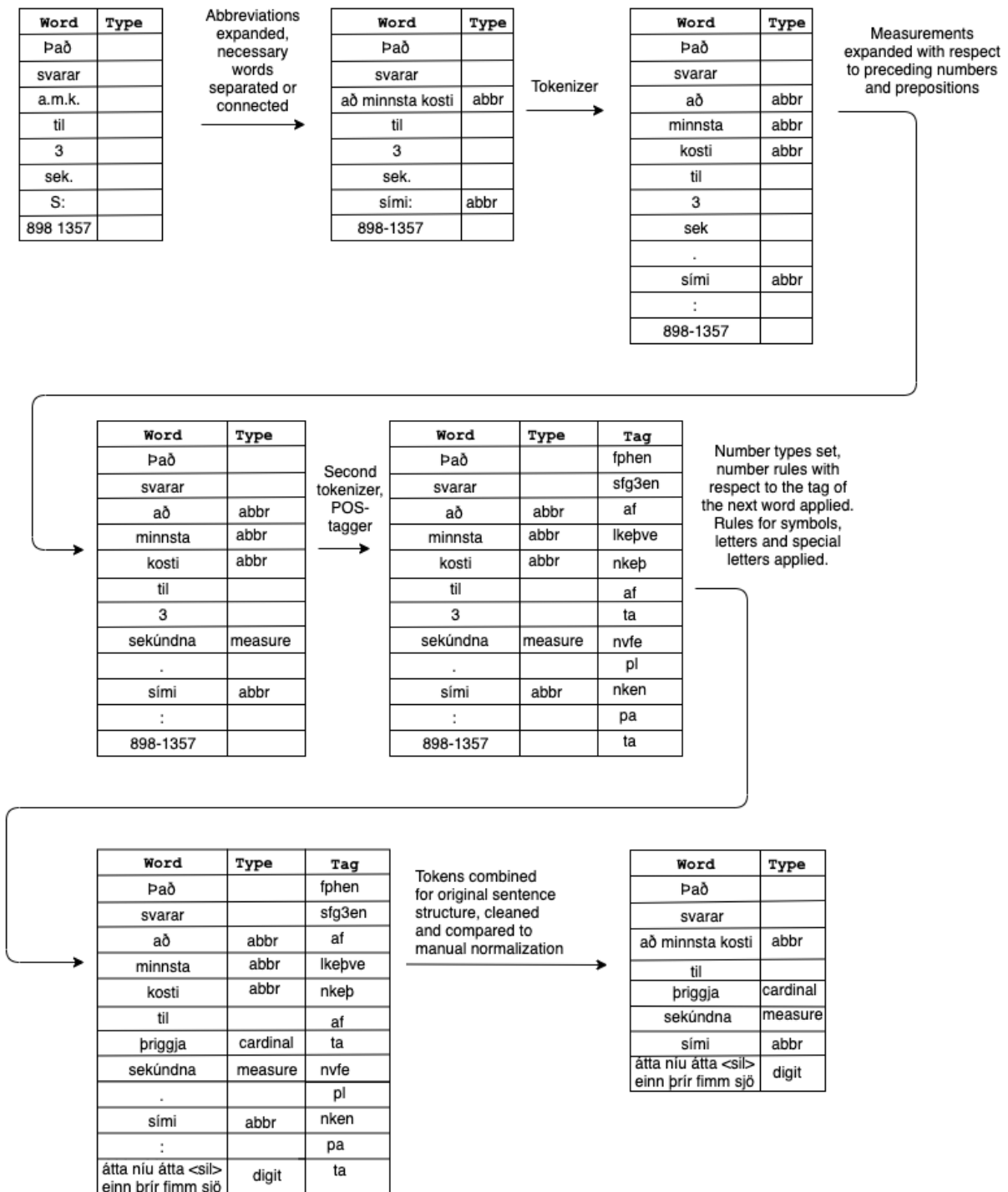


Figure 1: Pipeline of Regína from unnormalized to normalized text.

B Normalization guidelines

SEM. CLASS	EXPLANATION	EXAMPLE	NORMALIZED
PLAIN	Words remain same	dæmum	dæmum
PUNCT	Punctuation marks	„?!:;” „...	„?!:;” „...
CARDINAL	Cardinal numbers	86.761	áttatíu og sex þúsund sjö hundruð sextíu og einn/ein/eitt/eina/einum /einni/einu/eins/einnar
		337.429	þrjú hundruð þrjátíu og sjö þúsund fjögur hundruð tuttugu og níu
ORDINAL	Ordinal numbers	86.761.	áttatíu og sex þúsund sjö hundruð sextugasti og fyrsti / sextugasta og fyrsta/sextugustu og fyrstu
		337.429.	þrjú hundruð þrjátíu og sjö þúsund fjögur hundruð tuttugasti og níundi /tuttugasta og níunda/tuttugustu og níundu
LETTERS	Letter sequences	KR ehf	K R (ká err) E H F (e há eff)
DATE	Dates	1919 29. september 1928	nítján hundruð og nítján tuttugasti/a og níundi/a september nítján hundruð tuttugu og átta
		14. mars	fjórtándi/a mars
		september 2008	september tvö þúsund og átta
		kl. 20:00 klukkan 11.15	klukkan tuttugu núll núll klukkan ellefu fimmtán
MEASURE	Measurements	120 kW 5% 39,5 kg	hundrað og tuttugu kíló(vött/vöttum/vatta) fimm prósent(um/a) þrjátíu og níu komma fimm kíló(um/a)
SYMB	Symbols	+ - @ ©	plús mínus hjá höfundarréttur
ABBR	Abbreviations	a.m.k. SV-átt	að minnsta kosti suðvestanátt
WLINK	Web handles	helgas@ru.is @BarackObama #ljosanott2014	h e l g a s hj á r u punktur i s hjá B A R A C K O B A M A myllumerki l j o s a n o t t tveir núll einn fjórir
DECIMAL	Decimal numbers	0,45	núll komma fjórir/fjóra/fjörum /fjögurra/fjórar/fjögur fimm

SEM. CLASS	EXPLANATION	EXAMPLE	NORMALIZED
SPORT	Sports results	2-1 3:0 16/5 (fráköst)	tvö eitt þrjú núll sextán <sil> fimm (fráköst)
RNUM	Roman numerals	XII	tólf(ti/ta/tu)
FRACTION	Fractions	½ 2/6 1 1/3	hálfur/hálfan/hálfum/hálfs/hálf/hálfa /hálfri/hálfrar/hálft/hálft/hálfu tveir/tvo/tveimur/tveggja/tvær/tvö sjöttu einn og einn þriðji / einn og einn þriðja / einum og einum þriðja/eins og eins þriðja / ein og ein þriðja / eina og eina þriðju / einni og einni þriðju / einnar og einnar þriðju / eitt og eitt þriðja/einu og einu þriðja / eins og eins þriðja
DIGIT	Digit numbers	1109-05-420 365	einn einn núll níu <sil> núll fimm <sil> fjórir tveir núll þrír sex fimm
MONEY	Monetary amounts	3000 kr. kr. 4000 \$40 38 m.kr.	þrjú þúsund krónur/krónum/króna fjögur þúsund krónur/krónum/króna fjörutíu dollara(r/um) þrjátíu og átta milljón(ir/um/a) króna

B.1 Rules

- Separate a word that's built from letters and numbers, C19 becomes C nítján, 1.ferð becomes 1. ferð → fyrsta ferð.
- Delete a dash at the start of the line.
- If a word ends in dash it is ignored.
- @ is written *hjá*.
- = is written *jafnt og*.
- Links are written like *www.mbl.is/123* → *w w w punktur m b l punktur i s skástrik einn tveir þrír*, all letters are separated except for symbols and numbers, they are written out.
- For basketball results like *24/14 fráköst*, the / is written as <sil>, i.e., *24/14 fráköst* → *tuttugu og fjögur <sil> fjórtán fráköst*.
- In digit sequences, dashes are written as <sil>, e.g., *234-353-42* → *tveir þrír fjórir <sil> þrír fimm þrír <sil> fjórir tveir*

B.2 Ambiguities

- **DASH:** can imply *bandstrik (dash)* (links), <sil> (sports results), *til* (number intervals) or nothing.
- **SLASH:** can imply *skástrik (slash)* (links), *og (and, eða (or))*, a fraction, a <sil> or nothing.

The Danish Gigaword Corpus

Leon Derczynski

ITU Copenhagen
Denmark
ld@itu.dk

Manuel R. Ciosici

USC Information Sciences Institute
USA
manuelc@isi.edu

Rebekah Baglini

Aarhus University
Denmark

Morten H. Christiansen

Aarhus University & Cornell University
Denmark

Jacob Aarup Dalsgaard

Aarhus University
Denmark

Riccardo Fusaroli

Aarhus University
Denmark

Peter Juel Henriksen

Danish Language Council
Denmark

Rasmus Hvingelby

Alexandra Institute
Denmark

Andreas Kirkedal

ITU Copenhagen
Denmark

Alex Speed Kjeldsen

University of Copenhagen
Denmark

Claus Ladefoged

TV2 Regionerne
Denmark

Finn Årup Nielsen

Technical University of Denmark
Denmark

Jens Madsen

Karnov Group
Denmark

Malte Lau Petersen

Aarhus University
Denmark

Jonathan Hvithamar Rystrom

Aarhus University
Denmark

Daniel Varab

Novo Nordisk & ITU Copenhagen
Denmark

Abstract

Danish language technology has been hindered by a lack of broad-coverage corpora at the scale modern NLP prefers. This paper describes the Danish Gigaword Corpus, the result of a focused effort to provide a diverse and freely-available one billion word corpus of Danish text. The Danish Gigaword corpus covers a wide array of time periods, domains, speakers' socio-economic status, and Danish dialects.

1 Introduction

It is hard to develop good general-purpose language processing tools without a corpus that is broadly representative of the target language. Further, developing high-performance deep learning models

requires hundreds of millions of tokens (Radford et al., 2019; Raffel et al., 2020). To address this gap for Danish, a North Germanic/Scandinavian language spoken primarily in Denmark, we propose an open giga-word corpus. This corpus is free to download and use, thus enabling researchers and organizations to further develop Danish NLP without worrying about licensing fees. The corpus is a first necessary step to allow Danish speakers to receive the many benefits of the powerful range of NLP technologies.

This paper details the Danish Gigaword Corpus (DAGW), a billion-word corpus of language across various dimensions, including modality, time, setting, and place.

It is tricky to collect such a corpus automatically: automatic language identification tools confound closely related languages, especially Danish and

Bokmål, and are likely to miss important data (Radford et al., 2019; Haas and Derczynski, 2021). Existing representations underperform for Danish: the multilingual FastText embeddings (Joulin et al., 2018) miss core Danish words such as “træls”; Multilingual BERT lacks sufficient support for the Danish vowel “å”.¹

To remedy this situation, we propose a Danish Gigaword Corpus. The overriding goals are to create a dataset that is (1) representative, (2) accessible, and (3) a general-purpose corpus for Danish.

2 Background

Today’s NLP is generally data-intensive, meaning that large representative corpora tend to correlate with better models and better processing results. However, large representative corpora are available for only a small set of languages; there are fewer than ten manually-compiled gigaword-scale corpora, for example, and none for Danish.

Several substantial Danish text corpora have been compiled during recent decades. CLARIN-DK offers a variety of individual corpora of varying genres, annotations, and writing times. However, non-commercial licensing restricts corpus usage. Some major Danish corpora are related to dictionary production, as is the case for the 56 million words Korpus-DK available for search at the dictionary site ordnet.dk.² Leipzig Corpora Collection assembles Danish corpora from the Web, news sites, and Wikipedia (Goldhahn et al., 2012). The combined size of these corpora is orders of magnitude smaller than The Danish Gigaword Corpus. By themselves, these corpora do not meet the data size needs of modern language models.

Modern language models like T5 (Raffel et al., 2020) and GPT2 (Radford et al., 2019) are text-hungry, making automatic corpora construction attractive. Massive, monolithic, automatically collected datasets of web content, such as Common Crawl, support the training of large language models but suffer from quality issues (Radford et al., 2019) and bias (Ferrer et al., 2021). Models trained exclusively with such data quickly delve into generating toxic language (Gehman et al., 2020). Fur-

¹BotXO maintains a Danish BERT instance at https://github.com/botxo/nordic_bert.

This model was trained exclusively on uncurated web text and, therefore, (a) has a spurious understanding of Danish among other languages and (b) is particularly susceptible to the kind of toxic language identified by Gehman et al. (2020).

²<http://ordnet.dk>

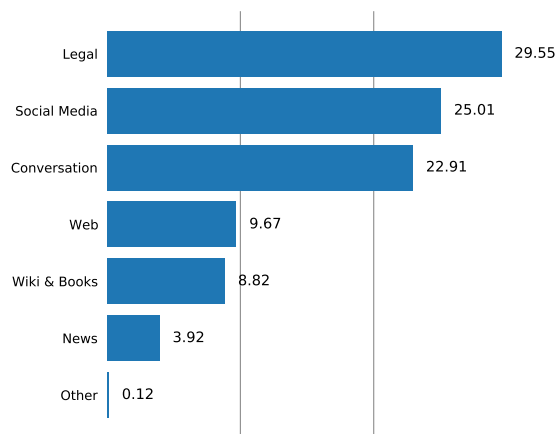


Figure 1: Content by domain (% of corpus).

thermore, the Danish section of Common Crawl is plagued by significant amounts of non-Danish content, in part due to the pervasive confusion between Danish and Norwegian Bokmål by highly multilingual language ID classifiers (Haas and Derczynski, 2021). Datasets derived exclusively from Common Crawl also have a bias toward webspeak and content from recent years, leaving models built over them sub-optimally prepared to process older Danish.

The lack of a large and qualitative Danish corpus causes Danish NLP tools to lag behind equivalent tools for better-resourced languages, and the gap is increasing (Pedersen et al., 2012; Kirkedal et al., 2019; Kirchmeier et al., 2020).

The first gigaword corpus was the English Gigaword (Graff et al., 2003), consisting of roughly one billion (10^9) words of English-language newswire text. The content was single-genre, national and global newswire, published between 1994 and 2002. Other gigaword corpora emerged later, for French, Arabic, Chinese, and Spanish. Even Icelandic, a language with just over 360 000 speakers, has a healthy gigaword project (Steingrímsson et al., 2018).

3 Linguistic diversity

For a corpus to be useful for a wide range of applications, it must include a wide range of language, mixing domains, speakers, and styles (Biber, 1993). Failing to do this can lead to severe deficiencies in the data. For example, when NLP work started on social media text, the Wall Street Journal-trained part of speech taggers missed essential words such as “Internet” (due to the articles being from the late

eighties and early nineties) and “bake”, due to their domain.

Common Crawl’s undirected collection of content often over-represents some dialects at the expense of other dialects. GeoWAC (Dunn and Adams, 2020) uses demographic information to construct English corpora that balance dialects. Unfortunately, a demographic- and Web-based approach underrepresents Danish dialects such as the endangered Bornholmsk dialect (Mortensen, 2016), which is almost absent from the Web.

These deficiencies do not form a solid basis for general-purpose NLP. So the Danish Gigaword Corpus captures and distributes as broad a range of Danish language use as possible, explicitly including language from a variety of settings (long-form writing, novels, social media, speeches, spontaneous speech), domains (news, politics, fiction, health, social media, law, finance), time periods (from the 1700s to present day), registers (formal, informal), and dialects (including, e.g., Bornholmsk and Sønderjysk).

4 Dataset construction

The Danish Gigaword Corpus consists of sections, with each section corresponding to a single source of text. Following prior efforts to construct broad-coverage datasets (Derczynski et al., 2016), sections are selected based on how well they help the corpus’ coverage of Danish language use over a variety of dimensions, including: time of authorship; speech situation; modality; domain; register; age of utterer; dialect of utterer; socio-economic status of utterer. This is a strong, intentional departure from editions of English Gigaword that focused on newswire. Achieving some degree of representativeness (Biber, 1993) requires the inclusion of sources beyond newswire text. We provide an overview of The Danish Gigaword Corpus’s content in Figure 1 and detail the sections in Table 1 and the appendix.

The Danish Gigaword Corpus follows the definition of genre used by Biber (1993), grounded in “situationally defined categories”, such as a language style recognized by (or used to define) a community, such as news articles, personal letters, or online chat; a domain as a particular topical focus (or set of foci) that are discussed, such as biomedicine, politics, or gaming; and a medium as the means by which communication is conducted, such as writing, online chat, conversation, and so

on. There is a natural overlap between medium and speech situations, but the delineation is beyond this work’s scope.

While the goal of DAGW is to cover a range of genres, domains, and media, it is difficult to measure the prevalence of each of these across all Danish users, let alone then gather and redistribute this data. Therefore, the goal is to cover something of everything that can be feasibly included, without letting any particularly monolithic combination dominate (in contrast to, e.g., the 100% written newswire content of English Gigaword v1 or the 100% Common Crawl content of GeoWAC). Not every intersection between genres, domains, and media can be covered, nor represented proportionally, in the first version of this corpus. Table 1 contains an overview of the genres, domains, and modalities included in the Danish Gigaword Corpus.

4.1 Data and metadata unification

Each section is contained in one directory, named after the “prefix” for the section. Each file in a section represents a single UTF encoded document. Each section contains at least two functional files: one describing how the section is licensed and one describing metadata about each document. For multi-speaker corpus sections, an optional file can contain a dictionary keyed by speaker ID. This assumes speaker IDs are used consistently through all documents in that section. Appendix B contains a complete description of the file format.

Sections are managed individually as part of a larger repository of the whole Danish Gigaword Corpus. A validation script helps make sure that the sections comply with the file format.

4.2 Data protection

The corpus does not contain “sensitive” data as per the GDPR definition; that means no information identifying sexual orientation, political beliefs, religion, or health connected with utterer ID. This is achieved by stripping utterer information from social media content. Thus, data discussing potentially personally sensitive topics, for example, social media around political discussions, is disconnected from personally-identifying information. Further, social media content is supplied not as plain text but as IDs and code for rehydration, a process where the content is re-downloaded, thus avoiding redistribution of this content and affording

	Date	Form	Domain	Dialect	Socioeconomic status	Size (M)
Legal						308.8
Retsinformation	contemporary	written	Laws	legal	high	188.4
Skat.dk	contemporary	written	Tax code	legal	high	52.8
H-Sø	contemporary	written	Court cases	mixed	mixed	67.6
Social Media						261.4
Hestenettet	contemporary	written	forum	mixed	mixed	228.9
General Discussions	2019 - 2020	written	Twitter	mixed	mixed	32.0
Parliament Elections	2019	written	Twitter	mixed	mixed	0.5
Conversation						239.4
OpenSubtitles	contemporary	spoken	Movie subtitles	mixed	mixed	130.1
Folketinget	2009 - 2019	spoken	Debates	rigsdansk	high	60.6
Europarl	2004 - 2008	spoken	Debates	standard	mixed	47.8
Spontaneous speech	2019	spoken	Conversation	mixed	mixed	0.7
NAAT	1930 - now	spoken	Speeches	rigsdansk	high	0.2
Web						101.0
Common Crawl	contemporary	written	Web	mixed	mixed	101.0
Wiki & Books						92.2
Wikipedia	2019 - 2020	written	Encyclopaedic	standard	mixed	55.6
Danish Literature	1700 - now	written	Literature	standard	mixed	25.6
Gutenberg	1700 - now	written	Literature	standard	mixed	3.2
WikiBooks	2019 - 2020	written	Manuals	standard	mixed	2.6
WikiSource	1700 - now	written	Literature	standard	mixed	2.5
Johannes V. Jensen	-	written	JVJ's works	rigsdansk	unknown	2.1
Religious texts	-	written	Religious	rigsdansk	unknown	0.6
News						40.0
TV2R	2015 - 2019	written	News	rigsdansk	high	10.0
DanAvis	1999 - 2003	written	News	rigsdansk	medium	30.0
Other						1.2
Dasem data ³	contemporary	written	Other	mixed	mixed	0.7
Botxt	contemporary	written	Other	Bornholmsk	mixed	0.4
DDT	contemporary	written	Other	mixed	mixed	0.1
Sønderjysk	contemporary	written	Sønderjysk	Sønderjysk	mixed	0.02
TOTAL						1045

Table 1: Text dimensions by text source in the Danish Gigaword corpus. Size in millions of words.

social media users the ability to delete their content without it being preserved by Danish Gigaword.

4.3 Test/Train partitions

Following the result that fixed test/train splits lead to unreliable results (Gorman and Bedrick, 2019), we avoid setting explicit test/train partitions in Danish Gigaword. We encourage users to select multiple random test splits. Since the Danish Gigaword is highly diverse, selecting multiple random splits will result in test sets with different biases following best practices (Søgaard et al., 2021).

4.4 Licensing

All corpus parts are licensed openly, for free distribution. We implement this with a mixture of Creative Commons general license (CC0) and CC-BY.

Some older corpora (e.g., Kromann et al. (2003)) used the right under Danish copyright law to cite small excerpts of up to 250 words from published articles. While this is a creative solution to sharing digital language data, Danish Gigaword uses almost exclusively whole articles, as they are easier to work with, providing full context.

5 Distribution and sustainability

As mentioned earlier in this paper and by Kirkedal et al. (2019); Kirchmeier et al. (2019, 2020), one problem that plagues Danish NLP is a lack of large accessible corpora. To address this and maintain strict licensing standards that permit open and free redistribution, Danish Gigaword Corpus is hosted and freely distributed via <https://gigaword.dk/>. Alternative downloads will be provided through

major dataset distribution services at each significant release.

DAGW is an intrinsically open project. In a bid to improve and uphold its relevance at a broad level, the current group of participants covers academia, industry, and the public sector. However, the DAGW project is also volunteer-led and volunteer-driven, which brings intrinsic risk. Aside from cross-sector involvement, the DAGW project attempts to mitigate that risk through licensing, distribution, membership, community, and data integrity policies.

Strategically, the corpus strives for an improved balance. The contents in the first release, with this paper, reflect the data that is available in Denmark. Data that is legally required to be open and unlicensed dominates the corpus, reflecting the current state of text sharing in Denmark. We hope that this will become less conservative over time and particularly look forward to further donations of newswire and literature, so that NLP for Danish can start to offer Danish speakers improved technology.

The data is licensed CC-BY and CC0, which gives it broad reach and applicability, and makes it easier for stakeholders to join than copyleft or non-commercial licenses, such as GPL or CC-NC, would. It also improves distribution prospects: because of this licensing choice, DAGW can be hosted at a third-party research data repository like Zenodo or Figshare, shifting the responsibility for data hosting and provision to specialized third parties. The DAGW project also maintains an open policy, with any qualified stakeholders welcome to join, especially if there is a compatible donation of data. Denmark's size helps keep a manageable community. The Danish Gigaword also fosters community involvement by publishing results – for example, this paper. Finally, a small toolkit is included in the project's Github repository for automatic validation of any committed data, ensuring content integrity, quality, and uniformity.

6 Conclusion and Future Work

In Denmark, natural language processing is nascent and growing faster and faster. Content restrictions and conservative licensing abound. This paper presents the Danish Gigaword Corpus, a unified effort across many institutions and many Danish speakers to construct a billion-word corpus representing the language. It aims to be useful to a maximally broad and diverse group of users.

The Danish Gigaword Corpus is an active project. There is continuing effort to add sources that enhance the corpus' breadth, including fiction, older works from the 1800s, and newswire. DAGW continues past the first billion words, with data always released under Creative Commons license and freely distributed via <https://gigaword.dk/>.

We hope that this concrete and significant contribution benefits anyone working with Danish NLP or performing other linguistic activities and encourages others to publish language resources openly.

Acknowledgments

This work was not supported by any funded project or university initiative, but rather was a labour of love by the first two, “*fremmedarbejder*”, “*tosprogede*” authors, who thought Denmark really ought to have a decent-sized open corpus of Danish. And now it has. We are extremely grateful for the generous contributions of time, effort, and data from so many that made this project possible.

References

- Douglas Biber. 1993. Representativeness in Corpus Design. *Literary and Linguistic Computing*, 8(4):243–257.
- Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395.
- Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad Twitter Corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179, Osaka, Japan. The COLING 2016 Organizing Committee.
- Leon Derczynski and Alex Speed Kjeldsen. 2019. Bornholmsk natural language processing: Resources and tools. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 338–344, Turku, Finland. Linköping University Electronic Press.
- Christina Dideriksen, Riccardo Fusaroli, Kristian Tylén, Mark Dingemanse, and Morten H Christiansen. 2019. Contextualizing conversational strategies: Backchannel, repair and linguistic alignment in spontaneous and task-oriented conversations. In *Proceedings of the 41st Annual Conference of the Cognitive Science Society*, pages 261–267. Cognitive Science Society.
- Jonathan Dunn and Ben Adams. 2020. Geographically-Balanced Gigaword Corpora for 50 Language Varieties. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2521–

- 2529, Marseille, France. European Language Resources Association.
- Xavier Ferrer, Tom van Nuenen, Jose M. Such, and Natalia Criado. 2021. Discovering and Categorising Language Biases in Reddit. In *Proceedings of the 15th International Conference on Web and Social Media*.
- Riccardo Fusaroli, Bahador Bahrami, Karsten Olsen, Andreas Roepstorff, Geraint Rees, Chris Frith, and Kristian Tylén. 2012. Coming to terms: Quantifying the benefits of linguistic coordination. *Psychological Science*, 23(8):931–939. PMID: 22810169.
- Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.
- Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).
- Kyle Gorman and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791, Florence, Italy. Association for Computational Linguistics.
- David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English Gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.
- René Haas and Leon Derczynski. 2021. Discriminating Between Similar Nordic Languages. In *Proceedings of the Workshop on NLP for Similar Languages, Varieties and Dialects*.
- Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. User review sites as a resource for large-scale sociolinguistic studies. In *Proceedings of the 24th International Conference on World Wide Web, WWW '15*, pages 452–461, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in Translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium. Association for Computational Linguistics.
- Sabine Kirchmeier, Peter Juel Henriksen, Philip Diderichsen, and Nanna Bøgebjerg Hansen. 2019. *Dansk sprogteknologi i verdensklasse*. The Danish Language Council.
- Sabine Kirchmeier, Bolette Pedersen, Sanni Nimb, Philip Diderichsen, and Peter Juel Henriksen. 2020. World class language technology - developing a language technology strategy for Danish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3297–3301, Marseille, France. European Language Resources Association.
- Andreas Kirkedal, Barbara Plank, Leon Derczynski, and Natalie Schluter. 2019. The Lacunae of Danish Natural Language Processing. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 356–362, Turku, Finland. Linköping University Electronic Press.
- Alex Speed Kjeldsen. 2019. Bornholmsk Ordbog, version 2.0. *Mål og Måle*, 40. årgang:22–31.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*, volume 5, pages 79–86.
- Matthias T Kromann, Line Mikkelsen, and Stine Kern Lyng. 2003. Danish Dependency Treebank. In *Proc. TLT*, pages 217–220.
- Anders Edelbo Lillie, Emil Refsgaard Middelboe, and Leon Derczynski. 2019. Joint rumour stance and veracity prediction. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 208–221, Turku, Finland. Linköping University Electronic Press.
- Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).
- Marianne Mortensen. 2016. Den bornholmske dialekt dør—og hvad så? Technical report, Roskilde Universitet.
- Bolette Sandford Pedersen, Jürgen Wedekind, Sabine Kirchmeier-Andersen, Sanni Nimb, Jens-Erik Rasmussen, Louise Bie Larsen, Steen Bøhm-Andersen, Peter Henriksen, Jens Otto Kjær, Peter Revsbech, Hanne Erdman Thomsen, Sanne Hoffensetz-Andersen, and Bente Maegaard. 2012. *Det danske sprog i den digitale tidsalder*. Springer.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Anders Søgaard, Sebastian Ebert, Joost Bastings, and Katja Filippova. 2021. We Need to Talk About Random Splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, Kiev, Ukraine. Association for Computational Linguistics.

Steinþór Steingrímsson, Sigrún Helgadóttir, Eiríkur Rögnvaldsson, Starkaður Barkarson, and Jón Guðnason. 2018. Risamálheild: A very large Icelandic text corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kristian Tylén, Riccardo Fusaroli, Pernille Smith, and Jakob Arnoldi. 2016. The social route to abstraction. *Cognitive Science*.

A Detailed corpus description

Here we detail some of the sections included in the corpus, specifying what they bring to the dataset to make it a rich resource covering a wide range of lexical, syntactic, and sociolinguistic phenomena expressed by Danish users. Table 1 provides an overview of the corpus.

A.1 TV2 Regionerne

This section is a contemporary Danish newswire sample: approximately 50 000 full newswire articles published between 2010 and 2019. It contains articles of regional interest, written following editorial standards. This section’s value is in both its temporal variation, covering a decade of events, and its spatial variation, covering many local events across most of Denmark (TV2 Bornholm is excluded). As a result of local event coverage, the section contains many locally relevant named entities, which might otherwise not be present in a dataset of national news.

A.2 Folketinget

The Danish parliament (Folketinget) keeps a record of all meetings in the parliament hall.⁴ All records have a transcript produced by commercial Automatic Speech Recognition (ASR) followed by post-editing by linguists employed by Folketinget for intelligibility, i.e., edit out dysfluencies, restarts, repairs, and mistakes. The transcript is, therefore, not a representation of spoken Danish but rather information content.

In the parliament hall, one speaker at a time addresses members of the parliament. Monologues

⁴There are no records of committee meetings or *samråd*.

may include rebuttals or other comments to statements in previous monologues. While speakers can read aloud from a prepared statement or speak extemporaneously, we expect no difference to be apparent in the data because of the post-editing.

The Folketinget section covers parliament hall sessions between 2009 and 2019. It contains discussions on a wide range of topics, issues, and named entities relevant to Danish society.

A.3 Retsinformation

The site retsinformation.dk provides access to Danish laws and regulations and documents from the Danish parliament (Folketinget). The text is provided by Folketinget, ministries, the ombudsman of Folketinget, and Rigsrevisionen. The legislative texts in this section include a variety of features: Uppercase text, redaction where names and addresses are left out, itemized text with chapter and section numbering, headlines, words with intra-letter spacing.

A.4 Spontaneous speech

The conversational corpus included originates from interdisciplinary research conducted within the Interacting Minds Center,⁵ and the Puzzle of Danish project⁶ at Aarhus University. Transcribed Danish speech is generally a rare kind of data, and spontaneous speech especially so; these manually transcribed conversations thus form a valuable resource. Spontaneous and pseudo-spontaneous conversations come from various contexts, e.g., getting to know each other, solving a puzzle together, or making joint decisions. The participants have agreed on releasing anonymized transcripts of their conversations. All conversations involve two speakers, sometimes conversing face-to-face, sometimes via a chat tool. Speech is transcribed post-hoc by native speakers. Studies published relying on this data include Fusaroli et al. (2012), Dideriksen et al. (2019), and Tylén et al. (2016).

A.5 Danish Wikipedia

This section comprises a dump of Danish Wikipedia⁷, stripped of Wikipedia-specific markup. The content is collaboratively written by a broad range of authors and covers many specific articles that often do not exist in other languages. Most

⁵<http://interactingminds.au.dk>

⁶<https://projects.au.dk/the-puzzle-of-danish/>

⁷<https://dumps.wikimedia.org/dawiki/>

content has been roughly checked for syntactic and orthographic canonicity by editors of the Danish Wikipedia and is a rich source of region-specific named entities, often situated in full, fluent sentences. The content is reproduced verbatim in accordance with the GNU Free Documentation License.

A.6 Europarl

The Europarl Parallel Corpus (Koehn, 2005) contains proceedings of the European Parliament in 21 European languages that were automatically extracted and aligned. We include the Danish part of the Europarl corpus and perform no pre-processing other than file format conversions.

A.7 OpenSubtitles

OpenSubtitles⁸ is a website where a community writes and shares subtitles for mostly big-budget movies. We extract the Danish subtitles from the OpenSubtitles section of OPUS (Lison and Tiedemann, 2016). We clean the corpus to fix issues such as the capital letter I instead of the lower case letter L. We remove files that do not contain any characters specific to Danish (i.e., any of the letters *å*, *æ*, or *ø*).

A.8 Religious text

This section contains a Danish translation of the Bible from the Massively Parallel Bible corpus (Christodouloupoulos and Steedman, 2015) without any pre-processing other than file format conversion. We continue to look for other sources of religious textual content to improve the coverage and significance of this section.

A.9 Danish Twitter

Social media content is rich in unedited text, allowing for a very broad range of expressions. We know that social media users typically vary their language use to afford some representation for what would typically be communicated non-verbally, and while there are corpora for this for e.g. English, there are very few published corpora containing Danish social media text (e.g., (Hovy et al., 2015; Lillie et al., 2019)). This section contains two datasets of Danish tweets as dehydrated content, and includes a script for rebuilding this part of the corpus, thus permitting GDPR-compliant redistribution. The first dataset contains approximately 29 000 tweets

⁸<https://www.opensubtitles.org>

in Danish from the #dkpol hashtag collected during the national parliamentary elections of 2019. The second dataset, consisting of approximately 1.6 million Danish tweets collected between April-June 2020, is not constrained by topic as tweets were collected using the 250 highest frequency Danish words.

A.10 DanAvis20

Corpus DanAvis20 consists of articles from various national Danish (daily) newspapers, including *Aktuelt*, *Berlingske Tidende*, *Dagen*, and *Weekendavisen*. The articles were published during 1999-2003. All texts included have been cleared for distribution under the CC0 license (cf. Section 4.4). As part of the clearing agreement, the papers were slightly edited by limiting all text quotes to 200 words (at most), picking sentences from longer papers at random. Sentences were mildly scrambled (DanAvis20 has no instances left of 4 adjacent sentences). Proper names were pseudonymized (except “Denmark”, “København”, “USA”, and a few others). Infrequent content words (10ppm or less) were replaced in situ by “statistical cognates”, i.e., words of similar frequency and equivalent morpho-syntactic form (e.g., replacing “Der er sardiner i køleskabet.” with “Der er skilsmissesager i forsikringsselskabet.” while keeping “Ministeren rejser hjem igen”). As overall statistical and lexical properties of DanAvis20 are thus kept invariant, the corpus still provides good material for most NLP training purposes.

A.11 The Bornholmsk Ordbog Dictionary Project

Fictional texts of various kinds written in Bornholmsk, the dialect spoken on the Danish island of Bornholm,⁹ have been digitized (OCR’ed and proofread) by volunteers working within the recently resumed *Bornholmsk Ordbog* dictionary project (Kjeldsen, 2019). Most of the material included is written by Otto J. Lund in the period 1930-48 (novels, short stories, and poems). The Bornholmsk subcorpus, which in its present state amounts to circa 400 K words, also includes folk stories published by J. P. Kuhre in 1938, and by K. M. Kofoed in 1935, fictional letters by various authors published in the 1930s, as well as poems by Alfred Jensen published in 1948 and various other

⁹The language code for Bornholmsk under IETF BCP-47 is da-bornholm.

texts from the same period. The non-standardized orthography varies considerably from source to source. The Bornholmsk part of the Danish Gigaword is a significantly extended dataset, well beyond that studied in earlier NLP work on the dialect (Derczynski and Kjeldsen, 2019).

B File format

The philosophy is to present data as plaintext, UTF8, one file per document. Accompanying metadata gives information about (for example) the author, the time or location of the document’s creation, an API hook for re-retrieval of the document, among others.

B.1 Corpus Sections

As the corpus many sections, per section, we do the following:

- Give each corpus section a directory with an agreed name.
- Keep all plaintext as one file per document.
- Use a section prefix, underscore, and document identifier as the filename, e.g., “tv2r_01672”.
- Do not use file extensions for the text files.
- Maintain a one-record-per-line JSONL file in the directory, with the same name as the section, and with “jsonl” suffix, e.g., “tv2r.jsonl”. The content of this file should follow the JSONL format, see <http://jsonlines.org>.
- Each document’s metadata is placed as a single JSON record in the JSONL metadata file, with a key “doc_id” matching the filename it describes. Separate entries by line breaks (i.e., one JSON object per line).
- A LICENSE file should be included in each section, stating the license under which the section is distributed. CC and public domain only! Preferably CC0 or CC-BY; CC-NC if we have to. No copyleft licenses - they restrict the use of the data too much, which we are trying to avoid.

Here are the fields for the standoff JSONL metadata file entries:

- `doc_id`: a string containing the document ID, which is also its filename. Begin with the section prefix, followed by an underscore. **String. Required.**

- `date_published`: the publication date of the source document, including the timezone. If only the year is available, use `year_published` instead. In the Python `strftime()` format, use “%c %z”. **String. Preferred.**
- `uri`: the URI from which the document originated; can be an API endpoint that links directly to the data. **String, URI. Preferred.**
- `year_published`: the year CE that the source document was published. **Integer.** Use only as an alternative to `date_published`. **Optional.**
- `date_collected`: the date at which the source document / API result collection, including the timezone. In the Python `strftime()` format, use “%c %z”. **String. Optional.**
- `date_built`: the date this document was included in the current version of the dataset, including the timezone. In the Python `strftime()` format, use “%c %z”. **String. Optional.**
- `location_name`: the name of the location of the document’s origin. **String. Optional.**
- `location_latlong`: latitude and longitude of the document’s origin. List of two floats. **Optional.**

B.2 Speech transcripts

To represent speakers in the text files, prefix each turn with “TALER 1:” (substituting whatever ID is appropriate). Note: there is no space before the colon; use one space after the colon. It is also OK to include the speaker’s name directly if this is publicly known, e.g., “Thomas Helmig:”.

For multi-speaker corpus sections, an optional `talere.jsonl` file can be included in the section, containing one JSON dictionary keyed by speaker ID. Speaker IDs should be consistent through all documents in a section. Speaker IDs need only be unique to speakers in a section, not universally.

DANFEVER: claim verification dataset for Danish

Jeppe Nørregaard
IT University of Copenhagen
jeno@itu.dk

Leon Derczynski
IT University of Copenhagen
leod@itu.dk

Abstract

Automatic detection of false claims is a difficult task. Existing data to support this task has largely been limited to English. We present a dataset, DANFEVER, intended for claim verification in Danish. The dataset builds upon the task framing of the FEVER fact extraction and verification challenge. DANFEVER can be used for creating models for detecting mis- & disinformation in Danish as well as for verification in multilingual settings.

1 Introduction

The internet is rife with false and misleading information. Detection of misinformation and fact checking therefore presents a considerable task, spread over many languages (Derczynski et al., 2015; Wardle and Derakhshan, 2017; Zubiaga et al., 2018). One approach to this task is to break down information content into verifiable *claims*, which can subsequently be fact-checked by automated systems.

Automated fact checking can be framed as a machine learning task, where a model is trained to verify a claim. Applying machine learning requires training and validation data that is representative of the task and is annotated for the desired behaviour. A model should then attempt to generalise over the labeled data.

One dataset supporting automatic verification is the Fact Extraction and VERification dataset (FEVER) in English (Thorne et al., 2018a), which supports the FEVER task (Thorne et al., 2018b; Thorne and Vlachos, 2019). The dataset is aimed both at claim detection and verification.

While the misinformation problem spans both geography and language, much work in the field has focused on English. There have been suggestions on strategies for alleviating the misinformation problem (Hellman and Wagnsson, 2017). It is

however evident that multilingual models are essential if automation is to assist in multilingual regions like Europe. A possible approach for multilingual verification is to use translation systems for existing methods (Dementieva and Panchenko, 2020), but relevant datasets in more languages are necessary for testing multilingual models' performance within each language, and ideally also for training.

This paper presents DANFEVER, a dataset and baseline for the FEVER task in Danish, a language with shortage of resources (Kirkedal et al., 2019). While DANFEVER enables improved automatic verification for Danish, an important task (Derczynski et al., 2019), it is also, to our knowledge, the first non-English dataset on the FEVER task, and so paves the way for multilingual fact verification systems. DANFEVER is openly available at https://figshare.com/articles/dataset/DanFEVER_claim_verification_dataset_for_Danish/14380970

2 English FEVER

The Fact Extraction and VERification dataset and task (FEVER) is aimed at automatic claim verification in English (Thorne et al., 2018a). When comparing we will stylize the original FEVER dataset ENFEVER to avoid confusion. The dataset was created by first sampling sentences from approximately 50,000 popular English Wikipedia pages. Human annotators were asked to generate sets of claims based on these sentences. Claims focus on the same entity as the sentence, but may not be contradictory to or not verifiable by the sentence. A second round of annotators labelled these claims, producing the labels seen in Table 1, using the following guidelines:

"If I was given only the selected sentences, do

I have strong reason to believe the claim is true (Supported) or stronger reason to believe the claim is false (Refuted)."

"The label NotEnoughInfo label was used if the claim could not be supported or refuted by any amount of information in Wikipedia."

The ENFEVER guidelines state that claims labelled NotEnoughInfo could possibly be verified using other publicly available information, which was not considered in the annotation.

Label	Verifiability	#	%
Supported	Verifiable	93,367	50.3
Refuted	Verifiable	43,107	23.2
NotEnoughInfo	NotVerifiable	48,971	26.4
Total		185,445	-

Table 1: Annotated classes in ENFEVER.

In the FEVER task (Thorne et al., 2018b), automatic verification is commonly framed as a two-step process: given a claim, relevant evidence must first be collected, and secondly be assessed as supporting or refuting the claim, or not providing enough information. ENFEVER contains data for training models for both steps.

We tasked annotators to create claims for DANFEVER based on the same guidelines and without regulation of class-distribution. The class-distribution of DANFEVER is therefore a bit different than that of ENFEVER; there is about the same ratio of Supported claims, but more Refuted and less NotEnoughInfo claims in DANFEVER than in ENFEVER.

3 Method

A FEVER task instance consists of a claim, zero or more pieces of evidence, and a label. The labels take one of the following values:

Supported Claims that can be supported by evidence from the textual data

Refuted Claims that can be refuted by evidence from the textual data

NotEnoughInfo Claims that can neither be supported or refuted based on the textual data

The claims were created based on data from Danish Wikipedia and Den Store Danske (a

privately-developed, non-profit, online encyclopedia based in Denmark and financed through foundations and universities). Both sites are generally considered high quality and trustworthy. Along with the claims, DANFEVER supplies the Wikipedia dump used for creating the claims as well as the content of the articles used from Den Store Danske. The remaining articles from Den Store Danske are not included (due to rights), and all articles should be considered to be iid. for modelling.

The format of the dataset can be found in Appendix A.1.

3.1 Dataset Goal

DANFEVER can be used for research and implementation of multi-lingual claim-detection. The dataset can be used for bench-marking models on a small language, as well as for fine-tuning when applying such models on Danish data.

3.2 Data Statement

The following is a data-statement as defined by Bender and Friedman (2018). The dataset consists of a text corpus and a set of annotated claims. The annotated part contains 6407 claims, with labels and information about what articles can be used to verify them.

Curation Rationale A dump of the Danish Wikipedia of 13 February 2020 was stored as well as the relevant articles from Den Store Danske (subset of site to adhere to rights). Two teams of two people independently sampled evidence, and created and annotated claims from these two sites (more detail in section 3.3).

Speaker Demographic Den Store Danske is written by professionals and is funded by various foundations for creating free information for the Danish public. Wikipedia is crowd-sourced and its writers are therefore difficult to specify, although the content is generally considered to be of high quality.

Annotator Demographic The annotators are native Danish speakers and masters students of IT.

Speech Situation The data is formal, written texts created with the purpose of informing a broad crowd of Danish speakers.

Language Variety and Text Characteristics The language of the texts is fairly formal Danish

<p>Claim 3152: “Udenrigsministeriet har eksisteret siden 1848.” <i>The Ministry of Foreign Affairs has existed since 1848.</i></p> <p>Evidence Extract: “Dette er en liste over ministre for Udenrigsministeriet siden oprettelsen af ministeriet i 1848.” <i>This is a list of ministers of the Ministry of Foreign Affairs since it was founded in 1848.</i></p> <p>Evidence Entities: wiki_93781</p> <p>Verifiable: Verifiable</p> <p>Label: Supported</p>
--

(a) A Supported claim.

<p>Claim 1306: “Hugh Hudson er født i England i 1935.” <i>Hugh Hudson was born in England in 1935.</i></p> <p>Evidence Extract: “Hugh Hudson (født 25. august 1936 i London, England) er en britisk filminstruktør.” <i>Hugh Hudson (born 25th of August 1936 in London, England) is a British film director.</i></p> <p>Evidence Entities: wiki_397805</p> <p>Verifiable: Verifiable</p> <p>Label: Refuted</p>
--

(b) A Refuted claim.

<p>Claim 2767: “Lau Lauritzen har instrueret både stumfilmen Skruerækkeren og vikingefilmen Når ræven flyver.” <i>Lau Lauritzen directed the silent film Skruerækkeren and the viking film Når Ræven Flyver.</i></p> <p>Evidence Extract: “”</p> <p>Evidence Entities: wiki_833896</p> <p>Verifiable: NotVerifiable</p> <p>Label: NotEnoughInfo</p>

(c) A NotEnoughInfo claim.

Table 2: Examples of claims. English translations are in *italic*.

from encyclopedias. It is considered to be consistent. Any deviation from Danish language is largely due to topics on history from non-Danish regions.

3.3 Sampling and Annotation

The main text corpus was created by storing the Danish Wikipedia dump of the time as well as a subset of pages from Den Store Danske, selected from the annotation process. Two strategies were employed for gathering specific texts for claims. A selection of pages with well-known topics were selected from Wikipedia’s *starred* articles and Den Store Danske (similar to the “popular articles” selection in ENFEVER). Furthermore a random selection of Wikipedia entities with abstracts were

Label	Verifiability	#	%
Supported	Verifiable	3,124	48.8
Refuted	Verifiable	2,156	33.6
NotEnoughInfo	NotVerifiable	1,127	17.6
Total		6,407	-

Table 3: Annotated classes in DANFEVER.

	Median	Mean	SD
Claims			
# Characters	45	50.18	22.02
# Tokens	7	8.46	3.86
# Evidence Entities	1	1.10	0.34
Evidence Extracts			
# Characters	260	305.56	257.20
# Tokens	47	53.75	44.64

Table 4: Claims and evidence extracts in dataset.

selected to ensure broad spectrum of topics. Random substrings were selected and passed to annotators, who created claims based on each substring, as in ENFEVER. The claims focus on the same entity as the substring’s source document and may be supported by the text in the substring, but may also be refuted or unverifiable by the substring. It is up to the annotator to decide on what type of claim to aim for (although the final label of each claim is provided by the next annotator).

The set of claims were subsequently revisited by another annotator, who labelled the claim as Supported, Refuted or NotEnoughInfo, based on the original substring used to generate the claim. The majority of the claims (80%) are generated based on Wikipedia pages, while 20% were based on articles from Den Store Danske. Note that claims are independent of the source and could be verified using any text; while the FEVER format presents a list of articles where evidence is present, this list is not exhaustive, just as in the TREC and TAC challenges. The two annotating teams reported Fleiss κ -scores of 0.75 and 0.82 measured on a reduced subset. The remaining data was annotated by a single annotator.

4 Dataset Details & Analysis

DANFEVER consists of 6407 claims. We have included one example from each class in Tables 2a, 2b and 2c, and shown the label distribution in Table 3.

Table 4 summarizes the lengths of claims and evidence extracts, as well as the number of entities linked to the claims.

Location	#	Person	#	Organization	#
Finland	184	Donald Trump	110	Aalborg Universitet	11
Danmark	109	Winston Churchill	73	FN	11
Preussen	89	Hillary Clinton	44	DR	10
USA	80	Mary Wollstonecraft	36	Københavns Universitet	9
Chile	79	George W. Bush	24	Electronics Art	9
København	71	Frederik Den Store	16	FC Barcelona	9
Tyskland	64	Obama	15	Apollo Rejser	8
Israel	57	Eastwood	13	Bananarama	8
Norge	54	Jens	9	EU	8
Storbritannien	49	Grant Rodiek	8	MTV	7

Table 5: Most frequent entities and number of occurrences.

4.1 Named Entities in Claims

The entities mentioned frequently in a corpus can give insight into popular themes in the data. In this case, the topic of the claims is particularly relevant. We present an automatic survey of DAN-FEVER’s entities. Entities in claims were identified using the DaNLP NER tool (Hvingelby et al., 2020), which identifies location (LOC), person (PER), and organization (ORG) entities. Those most frequently named are shown in Table 5.¹

5 Baseline: Recognizing Textual Entailment

The FEVER task consists of verifying claims based on a text corpus. One common strategy is to split the task into three components (as in the original work (Thorne et al., 2018a))

1. Document Retrieval: Retrieve a useful subset of documents from the corpora, based on the claim.
2. Sentence Retrieval: Retrieve a useful subset of sentences from those documents, based on the claim.
3. Recognize Textual Entailment: Classify the claims as *Supported*, *Refuted* or *NotEnoughInfo*, based on the claim and the subset of sentences.

To provide baseline performance for future research to benchmark against, we trained a baseline model on the final task; recognizing textual entailment. Since there are no evidence extracts for the *NotVerifiable* samples, we apply the random-sampling method from the original EN-FEVER paper, where evidence is randomly assigned from the data to each of these samples. We trained classifiers on the resulting 3-class problem.

¹Interestingly the most mentioned location is Finland

The transformer based model BERT (Devlin et al., 2019) has shown promising performance for claim verification (Soleimani et al., 2020), and the team (DOMLIN) with highest FEVER-score in the FEVER2.0 competition used a BERT-based system (Thorne et al., 2019). Using the transformers repository from HuggingFace (Wolf et al., 2020) we test; mBERT (Feng et al., 2020) (tag: `bert-base-multilingual-cased`), XLM-RoBERTa Small and XLM-RoBERTa Large (Conneau et al., 2020; Liu et al., 2019) (tags: `xlm-roberta-base` and `xlm-roberta-large`), and the Danish NordicBERT (BotXO, 2019). We use BERT’s sentence-pair representation for claims and evidence extracts. The classification embedding is then passed to a single-hidden-layer, fully-connected neural network for prediction. We first train the prediction layer, while freezing the weights of the language model, and consecutively fine-tune them both. We do this in a 10-fold cross-validation scheme for the 4 models.

Table 6 shows weighted-mean F1-scores, training parameters and info about the models. XLM-RoBERTa Large performed best, followed by mBERT and then XLM-RoBERTa Small. NordicBERT performed surprisingly poor. The learning curve of NordicBERT flattened out quickly and nothing further was learned despite the high learning rate used. NordicBERT was trained for Masked-Language-Modelling, but we are unsure whether it was also trained for Next-Sentence-Prediction like BERT (or even Causal-Language-Modelling like RoBERTa). If not, this may explain the poor performance on this task, even when NordicBERT has shown promising results for other tasks.

For comparison the multi-layer perceptron and decomposable attention models from the EN-FEVER paper (Thorne et al., 2018a) maintained

Model	F1 Train	F1 Test	Params	Time	BS	Epochs	LR	WD	DR
mBERT	94.5%	85.0%	110M	14h, 10m	32	40	10^{-5}	10^{-6}	0.3
XLm-RoBERTa Small	78.8%	78.5%	270M	11h, 40m	32	40	10^{-5}	0	0
XLm-RoBERTa Large	98.5%	90.2%	550M	18h, 20m	8	20	$5 \cdot 10^{-6}$	0	0
NordicBERT	65.5%	65.5%	110M	6h, 40m	32	20	0.001	0.0	0.1

Table 6: Model Evaluations. F1 score is weighted-mean. Params: number of parameters in model. Time: total training & evaluation time using 1 NVIDIA Tesla V100 PCIe 32 GB card; RMSProp optimizer. BS: batch size. LR: maximum learning rate in single-round, cosine schedule w/ 10% warm-up.²WD: weight decay. DR: dropout rate.

		Predicted		
		NEI	R	S
True Class	NEI	1118	7	2
	R	6	1643	507
	S	4	441	2679

Table 7: Test-set confusion matrix of xlm-roberta-large classifier.

an F1 score of respectively 73% and 88% on the verification subtask. The comparable performance indicates that pretrained, multilingual, language models are useful for the task, especially considering that DANFEVER is small relative to ENFEVER. We show the collective test-set confusion matrix of xlm-roberta-large in table 7 and note that it is much easier to disregard the randomized evidence (classify NotEnoughInfo (NEI)), than it is to refute or support claims, which is to be expected.

6 Conclusion

We have presented a human-annotated dataset, DANFEVER, for claim verification in a new language; Danish. DANFEVER can be used for building Danish claim verification systems and for researching & building multilingual claim verification systems. To our knowledge DANFEVER is the first non-English FEVER dataset, and it is openly accessible³. Baseline results are presented over four models for the textual-entailment part of the FEVER-task.

²Available in Huggingface’s library: https://huggingface.co/transformers/main_classes/optimizer_schedules.html#transformers.get_cosine_schedule_with_warmup

³https://figshare.com/articles/dataset/DanFEVER_claim_verification_dataset_for_Danish/14380970

7 Acknowledgments

This research was supported by the Independent Danish Research Fund through the Verif-AI project grant. We are grateful to our annotators (Jespersen and Thygesen, 2020; Schulte and Binau, 2020).

References

- Emily M. Bender and Batya Friedman. 2018. Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science. *Transactions of the Association for Computational Linguistics*, 6:587–604.
- BotXO. 2019. NordicBERT. https://github.com/botxo/nordic_bert.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. *arXiv:1911.02116 [cs]*. XLM-R.
- D. Dementieva and A. Panchenko. 2020. Fake News Detection using Multilingual Evidence. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 775–776.
- Leon Derczynski, Torben Oskar Albert-Lindqvist, Marius Venø Bendsen, Nanna Inie, Jens Egholm Pedersen, and Viktor Due Pedersen. 2019. Misinformation on Twitter during the Danish national election: A case study. In *Proceedings of the conference for Truth and Trust Online*.
- Leon Derczynski, Kalina Bontcheva, Michal Lukasik, Thierry Declerck, Arno Scharl, Georgi Georgiev, Petya Osenova, Toms Pariente Lobo, Anna Kolliakou, Robert Stewart, et al. 2015. Pheme: Computing veracity—the fourth challenge of big social data. In *Proceedings of the Extended Semantic Web Conference EU Project Networking session (ESCW-PN)*.
- Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *Naacl Hlt 2019 - 2019 Conference of*

- the North American Chapter of the Association for Computational Linguistics: *Human Language Technologies - Proceedings of the Conference*, 1:4171–4186. ISBN: 9781950737130 Publisher: Association for Computational Linguistics (ACL).
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT Sentence Embedding. *arXiv:2007.01852 [cs]*. ArXiv: 2007.01852.
- Maria Hellman and Charlotte Wagnsson. 2017. How can European states respond to Russian information warfare? An analytical framework. *European Security*, 26(2):153–170.
- Rasmus Hvingelby, Amalie Brogaard Pauli, Maria Barrett, Christina Rosted, Lasse Malm Lidegaard, and Anders Sjøgaard. 2020. DaNE: A Named Entity Resource for Danish. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4597–4604.
- Sidsel Latsch Jespersen and Mikkel Ekenberg Thygesen. 2020. Fact Extraction and Verification in Danish. Master’s thesis, IT University of Copenhagen.
- Andreas Kirkedal, Barbara Plank, Leon Derczynski, and Natalie Schluter. 2019. The Lacunae of Danish Natural Language Processing. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 356–362.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*. RoBERTa.
- Henri Schulte and Julie Christine Binou. 2020. Danish Fact Verification: An End-to-End Machine Learning System for Automatic Fact-Checking of Danish Textual Claims. Master’s thesis, IT University of Copenhagen.
- Amir Soleimani, Christof Monz, and Marcel Worring. 2020. BERT for Evidence Retrieval and Claim Verification. In *Advances in Information Retrieval*, pages 359–366, Cham. Springer International Publishing.
- The SQLite Consortium. 2000. SQLite. www.sqlite.org.
- James Thorne and Andreas Vlachos. 2019. Adversarial attacks against Fact Extraction and VERification. *arXiv:1903.05543 [cs]*. ArXiv: 1903.05543.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. FEVER: A large-scale dataset for fact extraction and verification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 809–819.
- James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The Fact Extraction and VERification (FEVER) Shared Task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.
- James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. The Second Fact Extraction and VERification (FEVER2.0) Shared Task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6, Hong Kong, China. Association for Computational Linguistics.
- Claire Wardle and Hossein Derakhshan. 2017. Information disorder: Toward an interdisciplinary framework for research and policy making. *Council of Europe report*, 27.
- Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.
- Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–36.

A Appendices

A.1 Format

DANFEVER contains three sqlite databases (SQLite Consortium, 2000); `da_fever.db`, `da_wikipedia.db` and `den_store_danske.db`.

The databases `da_wikipedia.db` and `den_store_danske.db` contain article data from Danish Wikipedia and Den Store Danske respectively. They contain an `id`-field, which is a numerical ID of the article (the `curid` for Wikipedia and a simple enumeration for Den Store Danske). They also contain the `text` and `title` of each article, as well as the `url` to that article.

The `da_fever.db` database contain the annotated claims. Each row in the database contain a claim and a unique `id`. With each claims comes the labels `verifiable` (Verifiable and NotVerifiable) and `label` (Supported, Refuted and NotEnoughInfo). The evidence column contain information about what articles were used to create and annotate the claim, and is composed by a comma-separated string, with IDs referring to the articles. The ID-format is `Y_X` where `Y` is either `wiki` or `dsd` to indicate whether the article comes from Danish Wikipedia or Den Store Danske, and `X` is the numerical id from that data-source. Finally the claims that were `Verifiable` contains an `evidence_extract` which is the text-snippet used to create and annotate the claim. Note that there may be some character-level incongruence between the original articles and the `evidence_extract`, due to formatting and scraping.

All three databases are also provided in TSV-format.

The data is publicly available at https://figshare.com/articles/dataset/DanFEVER_claim_verification_dataset_for_Danish/14380970

The Icelandic Word Web: A Language Technology Focused Redesign of a Lexicosemantic Database

Hjalti Daníelsson, Jón Hilmar Jónsson, Þórður Arnar Árnason, Alec Shaw,
Einar Freyr Sigurðsson, Steinþór Steingrímsson

The Árni Magnússon Institute for Icelandic Studies, Iceland

{hjalti.danielsson, jon.hilmar.jonsson, thordur.arnason, alec.shaw,
einar.freyr.sigurdsson, steinhor.steingrimsson}@arnastofnun.is

Abstract

The new Icelandic Word Web (IW) is a language technology focused redesign of a lexicosemantic database of semantically related entries. The IW's entities, relations, metadata and categorization scheme have all been implemented from scratch in two systems, OntoLex and SKOS. After certain adjustments were made to OntoLex and SKOS interoperability, it was also possible to implement specific IW features that, while potentially nonstandard, form an integral part of the Word Web's lexicosemantic functionality. Also new in this implementation are access to a larger amount of linguistic data, a greater variety of search options, the possibility of automated processing, and the ability to conduct research through SPARQL without possessing a mastery of Icelandic.

1 Introduction

We introduce the new Icelandic Word Web (IW; Icel. *Íslenskt orðanet*), a language technology focused overhaul and redesign of a lexicosemantic database of semantically related Icelandic words and phrases (Jónsson, 2017). This modernization improves access to the IW's intricate systems, makes its data more malleable, enables the use of a greater variety of metadata, and allows for a new, open-ended approach to conducting research on its various elements.

The IW is the only database of its kind for the Icelandic language. Although there does exist a number of other semantic databases, e.g. Arabic WordNet (Black et al., 2006), BalkaNet (Tufis et al., 2004), EuroWordNet (Vossen, 1998), IndoWordNet (Bhattacharyya, 2010), and The Multi-WordNet Project (Pianta et al., 2002), there is a strong tendency for these to be modeled on the

Princeton WordNet (Princeton University, 2010), arguably one of the best known databases of semantic word relations. While comparisons might be made between the IW and the Princeton WordNet, the IW diverges considerably in its overall structure and approach to semantic relations; its structure is more fluid and its focus more on the relations between core entries rather than the complex hierarchy around them (Rögnvaldsson, 2018). The implementation of the new IW itself represents a novel application of the two models with which the IW is encoded, and may prove useful in other projects involving the encoding of lexical databases with nonstandard structures and elements.

We begin by describing the core structure of the original IW, focusing on the aspects that remained unaltered. We then move on to the details of the overhaul. We discuss the choice of implementation models, how we applied them to the IW and what benefits we derived, and how we adapted them to certain aspects of the IW that were vital to its design but could not be represented by standard model features. We subsequently describe how the redesign has increased search scope, both in terms of the amount of accessible data and of the ways in which that data may now be searched for and inspected. Lastly, we touch on the potential future development and use of the IW, now that it is in this new form.

2 Core Structure of the Icelandic Word Web

The IW is effectively composed of two separate but interconnected systems: Entries and categories.

The former, entries, contains the words themselves and their semantic relations, and forms the bulk of the IW. Entries come in many varieties: Monolexical and polylexical, unordered and ordered (including phrasemes), sourced both from

reference works and primary sources, and accompanied by varying degrees of explanatory and morphosyntactic metadata (Jónsson, 2018). As is common with these types of collections, the entries do not have definitions except in cases where glosses are necessary to differentiate word forms; rather, their meanings are considered to be implicit in the relations they have to other entries or to their respective categories. The semantic relations themselves are similarly sourced both from primary sources and older reference works, with the majority being derived from the former.

The latter system, categories, contains a semantic classification scheme, and effectively functions as an ontology for the IW's entries. Unlike the entries and their relations, which are primarily derived from source material, the categories have been created and implemented over the years by the IW's past administrators. The scheme is descriptive rather than prescriptive, and is not intended to be all-encompassing; each entry may thus belong to one, none, or multiple categories. All categories have equal priority, there are no category hierarchies, and from a semantic perspective their subjects may overlap. Although categories do exist as separate entities in the IW, there are no direct category-to-category relations. They are connected only through the relations of the entities that belong to them.

The IW's primary type of semantic relation is a specific kind of parallel construction, which we will call *Pairings* for short. This relation indicates that two given entries, X and Y, have at some point appeared in a source text with the conjunction *og* (Eng. *and*) between them. Pairings are unordered by design, with a sourced *X og Y* being considered the equivalent of *Y og X*. Most of the other relation types in the original IW build in some way on Pairings, aside from a relatively small set of synonyms and antonyms whose handcrafted relations are drawn from preexisting entries in the IW's database. Pairings combine semantics and syntax, albeit with an emphasis on the former; and this amalgamated nature, coupled with their status as a cornerstone of the IW's full span of relation types, was a major design factor in the development of the new IW.

The creation of the original IW involved the work of several people, over a period of decades rather than years, collating relational information that initially described syntax and morphology but

later shifted in focus to involve semantics as well, all of which culminated in a deep and complex collection of data. While the original IW is presented through a web interface¹, there is no single, fully standardized type of entry in its underlying database. Some entries may be written or encoded differently from others, some have more metadata, and in certain cases the metadata itself may also be encoded differently between entries.

In short, a direct conversion to an established format was not an option. The only way of bringing the IW to a language technology friendly format while simultaneously maintaining its breadth of data and functionality was to design and implement it in the new format almost from scratch – a process that not only allowed for greater standardization, but also for greater inclusivity of information that up until now had either been difficult to use or entirely inaccessible.

3 General Implementation

Given that the IW's original structure is divided into two separate systems whose functionality is not the same, we approached the reimplementa-tion from the very beginning with the idea that it would not necessarily contain only one system. We therefore expected – and later found – one of the major points of complexity not to lie in how to fit the entire structure into one paradigm, but rather how well two new schemes could interact.

We modeled the new IW using two separate systems: OntoLex (McCrae et al., 2017) and SKOS (Miles and Bechhofer). Not only was each of these well suited to represent its respective part of the IW, but their point of intersection also turned out to be both fully workable – OntoLex's functionality has been designed with SKOS's interoperability in mind, so the two models mesh well when applied to the IW – and useful to model certain important and nonstandard aspects of the IW. Moreover, the use of these systems opened up the possibility of a range of new queries into the data that had not been possible until now, both through the new systemized encoding of its metadata, which made it available to users for the first time, and through the option of user-created SPARQL queries rather than fixed web site user patterns.

OntoLex was used to encode the IW's basic entities: Lemmas, their semantic relations, and per-

¹<https://ordanet.arnastofnun.is/>

tinent morphosyntactic information. It supports complex linguistic modeling, and was the model of choice for structuring an RDF version of the Princeton WordNet. It has already been used to recreate dictionaries in such a way that they are easily integrable with certain outside resources, an important point for the IW’s two systems. Moreover, it is the only data model of its kind that can reasonably be applied to a morphologically rich language such as Icelandic (Cimiano et al., 2016). OntoLex allowed us to recreate with relative ease a myriad of the original IW’s features, and, moreover, it enabled us to codify data that had been present in the original IW but had not been directly available to the user. As an example, most of the new IW’s monolexical entries are now accompanied by data drawn (when available and applicable) from the Database of Icelandic Morphology (DIM) (Bjarnadóttir et al., 2019). The data cover not only each entry’s lemmatized form but also its various inflectional forms and associated morphosyntactic features as well. This enables users to conduct both context-based searches for inflectional forms, and searches based on the morphosyntactic features themselves. These include gender, case, number, voice, mood, tense, person, and definiteness.

Additionally, certain entries (both mono- and polylexical) are labeled as exclamations, conjunctions, prepositions, numerals, set phrases and proper nouns, to the degree that the IW’s original data allows. The encoded data for polylexical entries is somewhat more sparse than for monolexical ones, partly to save space and avoid reduplicating data. However, a valuable new feature of the IW is interlinking: Wherever possible we have added to each single word of a polylexical entry a link to that word’s corresponding monolexical entry. This new feature grants access by proxy to the word’s morphosyntactic data, removing the need to reduplicate all that information in the polylexical entry, and overall greatly increases both the accessibility and interrelatedness of the IW’s data.

SKOS, a popular RDF-based ontology model, was used to encode the IW’s categories. While OntoLex has a great variety of various encoding options but a stringently ordered design dictating their use, the base version of SKOS has a comparatively smaller range of options but is far more malleable, a fact that makes it well-suited for the IW. Instead of being a standalone entity with a com-

plete and systematic internal structure, the IW’s category system draws heavily on the content of its other system of entities – categories are only created and put into use if existing lemmas support them – and there is a great deal of commingling and cross-referencing between the categories and lemmas, which means that the category system needs to be represented by a model that does not require all its entities to be discrete. Using SKOS, modeling the categories was a straightforward process. Where SKOS really comes into play is at the point where the IW’s two systems – and hence these two models – intersect.

4 Model Convergence

While OntoLex is an ideal option for representing the IW’s complex grammar, it is unable to encapsulate certain other aspects of the IW’s design. Most notably, OntoLex cannot comfortably represent semantic relationships such as the Pairings noted in the previous section, nor can it encode entities at all if, as is sometimes the case in the IW, they do not have an ontological connection to at least one category. SKOS, meanwhile, may be used to implement both these features but cannot store the entities themselves, which must be kept in OntoLex if we are to hold on to that linguistic modeling mentioned earlier.

Our solution was to develop a separate conceptual layer that hovers between these two models and serves as an intermediary. The change can be seen in Figures 1 and 2 below.

OntoLex			
Lexical Entry	Lemma A	Lemma B	Lemma C
Lexical Sense	↓	X	↓
SKOS Concept	Concept A	[No Concept]	Concept C
SKOS			

Figure 1: Potential IW implementation, without adjustment.

Figure 1 shows what the IW would be like if implemented directly in OntoLex and SKOS, without taking into account the aforementioned issues. The IW’s entities would be encoded in OntoLex as Lexical Entries, while its categories would be encoded in SKOS as Concepts. The two would then be connected by an OntoLex relation called

Lexical Sense. Entities that did not belong to an IW category would not have this kind of relation, which would render them invalid in OntoLex. Moreover, the IW's Pairings relation would need to be directly between Lexical Senses, and although OntoLex does support a number of sense-to-sense relations, none of them are a suitable fit for our purpose.

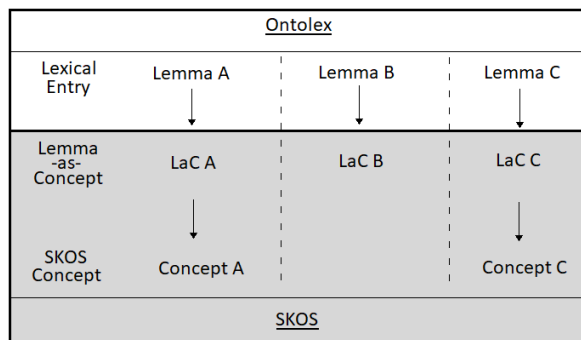


Figure 2: The new IW's actual implementation in OntoLex and SKOS.

Figure 2, on the other hand, shows our final implementation of the IW in OntoLex and SKOS, where these issues are taken into account. Here, we have added the separate conceptual layer. Note the replication of entities: After we have encoded every one of them in OntoLex as Lexical Entries, we *mirror* them in SKOS as Concepts. We call these mirrored entities *Lemma-as-Concept*, or LaC for short (*Lemma* being a more direct translation of the IW's Icelandic term for entities, *Fletta*) (Jónsson, 2017). From the viewpoint of OntoLex, LaCs serve as connectors to an ontology. From a SKOS viewpoint, LaCs are effectively a new category layer where each respective unit represents exactly one entity: The Lemma in question. In those cases where there does exist an actual category, the LaC merely functions as its subset.

Not only does this ensure that we always have the Lexical Entry/Concept connection mandatory for sustaining each Lexical Sense, but it also allows us to encode Lemma Pairings by using the SKOS *senseOf* keyword to relate LaCs as appropriate. In implementing this separate layer and its functionality, we have thus avoided creating nonstandard keywords that might have otherwise complicated the IW's use, and have confined any somewhat atypical use of the models to a clearly delineated section of our system, while simultaneously maintaining vital core functionality of the original IW. If new types of semantic word rela-

tions were to be added to the IW, they could comfortably be fitted into this layer.

5 Data Accessibility and Augmentation

In terms of existing entity relations, the new IW has greatly increased their scope. The original IW, which is accessible through a bespoke web site, contains one fundamental semantic relation – Pairings – and three ancillary ones (Synonyms, Near-Synonyms and Antonyms), plus a half-dozen derived relations that build on these. While these relations could often be highly informative, both in content and presentation, the precise nature of each relation was fixed and could not be altered by the user. Search functionality, likewise, was simple and clean but unmalleable, with a focus on textual searches for entities. Parameters both for the search and the relations themselves could not be altered, and the results could not be exported for further examination. The original data was stored in multiple tables in a database backend behind the web site, and was either not accessible except through the web site's search options or, in the case of morphosyntactic metadata, not accessible to regular users in any way.

The new IW, by contrast, effectively offers a limitless variety of potential searches. It is stored in a single RDF file accessible directly through CLARIN² under a CC BY 4.0 license. We have encoded only the Pairings and ancillary relations into the system, and the derived relations are not formally encoded in the system.

Instead, everything may now be produced through queries written in SPARQL, and the raw data itself may be viewed at will as needed. The sample query in Figure 3 shows a search for all words and phrases whose written form, irrespective of grammatical categorization, has more than one “gloss”, or explicitly mentioned definition. The SPARQL code is on the left, and the results on the right, with the results' left-hand column listing Word Web entries and the right-hand one listing their corresponding glosses.

These queries extend to practically every attribute encoded into the IW, including all those listed in the chapter on general implementation. So long as the user can formulate their intent into a valid SPARQL query, it may be applied to the IW's data. (This includes LaCs, which may be trivially folded into SPARQL queries.)

²<http://hdl.handle.net/20.500.12537/69>

```

SELECT ?term ?usage1
WHERE {
  {?Lemma rdf:type ontolex:Word}
  UNION
  {?Lemma rdf:type ontolex:MultiWordExpression} .
  ?Lemma ontolex:sense ?Sense1 .
  ?Sense1 ontolex:usage ?usage1 .
  ?Lemma ontolex:sense ?Sense2 .
  ?Sense2 ontolex:usage ?usage2 .
  ?Lemma ontolex:canonicalForm ?Form .
  ?Form ontolex:writtenRep ?term .
  FILTER (?Sense1 = ?Sense2) .
}
ORDER BY ?term ?usage

```

"fiskipakkaður"@is	"ilát, tunna"@is
"fiskisáll"@is	"bátur, skip"@is
"fiskisáll"@is	"á, vatn, mið"@is
"fiskpakkaður"@is	"ilát, tunna"@is
"fiskrikur"@is	"á, vatn, mið"@is
"fisléttur"@is	"byrði; tæki"@is
"fitjaður"@is	"tá"@is
"fitlaus"@is	"tá"@is
"fitulitill"@is	"mjólk, ostur"@is
"fitumikill"@is	"fiskur"@is
"fiturikur"@is	"mjólk"@is
"fiturikur"@is	"matur, fæða, fóður"@is
"fituskertur"@is	"matvæli"@is
"fituskuður"@is	"fiskur"@is

Figure 3: Word Web SPARQL query and corresponding output.

6 Conclusion and Future Developments

By its encoding in a publicly accessible form, the new IW reduces the barriers to entry for anyone wishing to make use of its stores of information. It also encodes that information in such a way that far more of it is accessible and usable for research, while maintaining, wherever possible, an adherence to official standards that ensure the IW's functionality is well-documented and comparable to that of any other models encoded using those same standards. On those occasions where that adherence is not possible or practical, we have tried to ensure that non-standard use is properly documented, kept to a minimum, and contained within a specific, clearly-defined part of the IW.

The IW's data storage is kept current, with new information added on a regular basis. As noted earlier, updates of existing data will grant the IW even greater usability. In addition, the models in which the IW is encoded support a range of potential information such as bilingualism and phonetics that, although not currently a part of the IW, may now be added to a system designed to store and handle this kind of data.

Overall, the IW is a deep and extensive system that models varying degrees of semantic relations between single- and multiword lemmas, drawing its information not from third party schemas and design, but rather directly from first-party sources. There is ample reason to think that the IW will be relevant to any number of research projects in the future, particularly now that it has been redesigned and reimplemented with depth of reach and ease of access in mind.

Acknowledgements

This work is supported by the Language Technology Programme for Icelandic 2019–2023, funded

by the Icelandic government.

References

- Pushpak Bhattacharyya. 2010. IndoWordNet. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 3785–3792, Valletta, Malta.
- Kristín Bjarnadóttir, Kristín Ingibjörg Hlynsdóttir, and Steinþór Steingrímsson. 2019. DIM: The Database of Icelandic Morphology. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 146–154, Turku, Finland.
- William Black, Sabri Elkateb, Horacio Rodriguez, Musa Alkhalifa, Piek Vossen, Adam Pease, and Christiane Fellbaum. 2006. Introducing the Arabic WordNet project. In *Proceedings of the third International WordNet Conference (GWC-06)*, pages 295–299, Seogwipo, Korea.
- Philipp Cimiano, John P. McCrae, and Paul Buitelaar. 2016. Lexicon model for Ontologies: Community Report. *W3C Ontology-Lexicon Community Group*.
- Jón Hilmar Jónsson. 2017. Frá orðabók að orðaneti. In Ásta Svavarsdóttir, editor, *Bundið í orð*, pages 1–26. The Árni Magnússon Institute for Icelandic Studies, Reykjavík.
- Jón Hilmar Jónsson. 2018. Íslenskt orðanet: Tekstbasert kartlegging og presentasjon av leksikalske relasjonar. *Nordiske Studier i Leksikografi*, 14:1–17.
- John P. McCrae, Julia Bosque-Gil, Jorge Gracia, Paul Buitelaar, and Philipp Cimiano. 2017. The OntoLex-Lemon Model: Development and Applications. In *Proceedings of eLex 2017 conference*, pages 587–597, Leiden, the Netherlands.
- Alistair Miles and Sean Bechhofer. SKOS Simple Knowledge Organization System Reference. *W3C Recommendation, year=2009, publisher=World Wide Web Consortium*.
- Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. 2002. MultiWordNet: Developing an

aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet*, pages 293–302, Mysore, India.

Princeton University. 2010. <https://wordnet.princeton.edu/> About WordNet.

Eiríkur Rögnvaldsson. 2018. Íslenskt orðanet: a treasure for writers and word lovers. *LexicoNordica*, 25:313–328.

Dan Tufis, Dan Cristea, and Sofia Stamou. 2004. BalkaNet: Aims, Methods, Results and Perspectives. A General Overview. *Romanian Journal of Information Science and Technology*, 7(1–2):9–43.

Piek Vossen. 1998. Introduction to EuroWordNet. In *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*, pages 1–17. Springer.

Getting Hold of Villains and other Rogues

Manfred Klenner & Anne Göhring & Sophia Conrad

Department of Computational Linguistics

{klenner|goehring|conrad}@cl.uzh.ch

Abstract

In this paper, we introduce the first corpus specifying negative entities within sentences. We discuss indicators for their presence, namely particular verbs, but also the linguistic conditions when their prediction should be suppressed. We further show that a fine-tuned BERT-based baseline model outperforms an over-generating rule-based approach which is not aware of these further restrictions. If a perfect filter were applied, both would be on par.

1 Introduction

In online media including social media, the world is often conceptualized as being divided into beneficiaries and benefactors, victims and villains. For quite some time, the most interesting questions seem to have been: Who is to blame and who benefits most. In this work, we strive to create a dataset and a first model to answer the first of these questions, i.e. to identify the villains in texts. But what is a villain, anyway? Are we compelled to reveal our moral convictions in order to answer this question? A murderer, a cheater, a liar seem to be clear cases. But what about white lies and the cheating in a card game? We could introduce a severeness score in order to quantify the villainousness grade.

In this paper, we describe our annotation efforts to create a corpus of sentences that comprises at least one entity that realises a negative (semantic) role. The filler of a negative semantic role might be a person, organization etc. But it also might be an event or even a non-animate physical object. Especially in metonymic constructions, non-animate fillers are to be expected. Although we also have started to annotate the strength of negativity, in this short paper we focus on the language usage that gives rise to the assignment of negative roles per se. In the second part of the paper we discuss two models: a rule-based and a BERT-based one.

2 Phenomena to be considered

The goal of our annotation is to identify those entities of a sentence that occupy a negative semantic role. A number of constructions can be used to take a negative perspective on some entity. One can do it explicitly by a noun phrase (*the lies of the president*), a predicative construction (*he is a liar*) or by using a verb who implies a negative actor (*He vilifies the people*). In this paper, we focus on verbs. It turns out, though, that not every usage of such a verb assigns a negative role. Only if the situation at hand is factual, then a negative role actually indicates a villain (Klenner and Clematide, 2016). Also ambiguity has to be taken into account. We, thus, are talking about a probability distribution, depending on various grammatical parameters. Before we have a more detailed look at this grammatical means, please note that quite a couple of verbs do have negative roles especially at the actor position. We have identified about 400 for the German language (Klenner and Amsler, 2016). Among them are verbs that indicate a crime (e.g. to murder, to kill, to injure, to torture ...), but also verbs like to vilify, to rebuff, to lie, to cheat, to mock, to demoralize, to prejudice and so on. Most of the time, the subject of the verbs bears the negative role.

As we said, metonymic reference has to be taken into account. Besides classical cases of metonymy like *producer for product* (e.g. *Pynchon is hard to read*), we also consider all references to be metonymic when humans are involved, e.g. *This agreement destroys our hope*.

There are a number of grammatical means (see Fig. 1) that indicate non-factuality and thus block the assignment of negative roles.

In reported speech (1), the actor of the reported event (*China*) is blocked. Subjunctive mood (2) inhibits the inference (*agreement*) since nothing has happened, which is also true for future tense (3). For verbs that have a theme dependent nega-

1. reported speech: *He said China was responsible for the virus*
2. subjunctive mood: *This agreement would destroy our hope*
3. future tense: *He will deny his guilt*
4. pronoun underspecification: *They admit it*
5. modal verbs: *The UN must invade*
6. modal adverbs: *He possibly is lying*
7. negation: *He has never cheated the people*
8. conditional constructions: *If he lies, the people won't elect him*
9. reflexive usages: *He cheats himself*
10. different reading: *He hurts the deadline*

Figure 1: Inference Blocker

tive actor assignment (like *to admit a mere/serious mistake/crime*), an unresolved pronoun (4) blocks inference. Some modal verbs (5) prevent the assignment of negativity (*UN*) as do adverbs (6) like *possibly*. Negation (7) also acts as a plug for such inferences, as well conditional statements (8). We also argue that the reflexive use of these verbs is not indicating a negative actor (9). Harder to detect are cases where the right reading should suppress the assignment of a negative actor (10).

In traditional machine learning we would use the items from Fig. 1 as features. A rule-based system could try to use them as filters. In a Deep Learning scenario, e.g. a BERT-based model, we could hope that the fine-tuning process will be sufficient to learn the regularities.

3 Annotated Corpus

As source for sentences that might have a negative role, we selected 1300 sentences from two corpora¹. The first one is a German newspaper corpus called TuebaDZ (Telljohann et al., 2009) comprising more than 100,000 sentences (publically available) and the second one are Facebook posts of a German right-wing party (AfD) with more than 300,000

¹The annotated data is available, just contact the first author.

sentences². The AfD texts also contain offensive language. The TuebaDZ data, on the other hand, comes from a left-oriented newspaper. We deliberately have chosen two different world views in order to have a broader range of examples.

We generated the candidate sentences by the following procedure: we parsed the sentences with the German ParZu parser (Sennrich et al., 2009) and then for those sentences that had a verb from our lexicon, we extracted the predicate argument structures (as a preprocessing step of our rule-based system, see (Klenner et al., 2017)) from the dependency parse trees. Finally, we identified the agent position (ARG0) and suggested it to be a negative actor. Given

Unser Land wird von den Medien zerstört

which translates to *Our country is being destroyed by the media*, the predicate argument structure (as a formula in Predicate Logic) is *destroy(media,country)*. From this, *media* was extracted to denote a negative actor *negative_role_filler(media)*. The two annotators were presented with the full sentence and had to determine whether the suggested negative actor actually is one. Moreover, the strength of negativity had to be determined on the basis of a scale from 1 (low) up to 3 (high). A zero means false positive.

In the course of the annotation, we removed a couple of sentences, because no actor was found by the predication extractor. We ended up with 1260 sentences. 460 cases are false positives, i.e. the found actor was not a negative actor, exactly 800 were true positives. We had a closer look at the reasons for the false positives, i.e. how the criteria from Fig. 1 are distributed. Only 4 cases would need coreference resolution, 18 are errors based on negation, 19 cases were future tense, 19 reported speech, 38 subjunctive mood, 48 reflexive usage, 59 conditional forms and 162 were wrong readings. We also had a number of parsing errors, namely 93 (wrong candidate). From 460 cases of false positives, thus, 183 cases (39.78 %) can be blocked by a perfect filter, coreference, negation, parsing errors and cases of wrong readings due to ambiguity are out of reach.

Our inter-annotator agreement is a Kappa score of 0.78: i.e. whether the annotators agreed that a noun candidate was really a negative actor or a false positive.

²The data is publically available on request - please contact the first author.

4 Experiments

4.1 Rule-based Baseline

We used our rule-based system for sentiment inference³ as a baseline. The system is verb-based and is designed to generate all pro (in favor of) and con (against) relations among the entities mentioned in a text. Moreover it indicates which discourse referents are negative actors and which receive a negative effect (see (Klenner et al., 2017) for the details). We just took the negative actors from the output and tested against our gold-standard. The system is over-generating: a rule triggers if a verb from the lexicon is found and the syntactic frame of the verb is met by the parse tree. We have not realised a filter (from Fig. 1) to block non-factual sentences from producing negative actors, but we give the hypothetical improvement the rule-based system would achieve if it was available (RB* in Tab. 1).



Figure 2: Top is right, bottom wrong

Figure 2 shows the result for the sentence *The contract destroys our hope* (top) and *The contract would destroy our hope* (subjunctive, bottom). The negative actor (*nac*) *Vertrag* (contract) stands in a con relation (red arc) with *Hoffnungen* (hopes) which receives a negative effect (*neff*). Only the top analysis is right, since the second sentence is not factual.

The advantage of such a rule-based system is transparency. The logic behind the predictions can be analysed, further refined, and applied to new verbs, if needed. The backside is that it remains brittle: lexical gaps (reduced recall) and erroneous parse trees (reduced precision) affect the performance. Moreover, if we are interested only in a well-performing system (in some end-to-end architecture), we do not necessarily need transparency. It might turn out that a neural approach is on par

³<https://pub.c1.uzh.ch/demo/stancer/>

with or even outperforms it. Also, the next step in our research strives to automatically quantify the severeness of negativity of an actor. Here a regression analysis is a natural approach (given that enough training material is available which is current work).

4.2 BERT-based Model

We realised a straightforward neural model by fine-tuning a German BERT model⁴. Several runs with different test sets showed that the results only slightly vary.

We tried two scenarios. In the first one, the whole sentences together with the labels were given to the training procedure. The binary classification task then was to label the sentence either as 1 (true positive) or 0 (false positive). If 1, then we know that the candidate noun (ARG0, which is mostly the subject of the verb from our verb lexicon) is a negative actor, given 0, it is not. The results were not very promising. We achieved 61% precision and 52% recall. We, thus, stopped further experiments with this setting.

In the second and simpler setting, the training procedure just gets all words between ARG0 and its verb (including ARG0 and the verb). Due to German word order, it might be the other way round as well (every word between the verb and ARG0). Sometimes, a potential indicator word (e.g. an adverb) gets lost in these cases, but they are rare. To give an example of such a fragment:

”er die Frauen immer wieder anschrie” (he yelled at the women again and again)

ARG0 is *er* (he) and the verb from the lexicon is *anschreien* (to yell). In Table 1 we provide the results of four runs with BERT (DL 1-4) and the single result from the application of the rule-based model (RB).

First of all, the RB model does not trigger on every sentence. The reason might be a missing verb subcategorization frame or a wrong dependency tree (the model only triggers if the verb frame from the lexicon is found). This explains the recall below 100%. The precision is lower than that of the DL model since the RB model is not able to identify class 0. Some of the predicted 1, thus, are 0. The recall of RB is higher than those of DL probably since no inference blocking mechanisms are imple-

⁴We use the Transformer library from HuggingFace (Wolf et al., 2020) and the BERT model made publicly available at <https://huggingface.co/bert-base-german-cased>

	precision	recall	f-measure
RB	64.87	88.04	74.83
RB*	71.83	88.04	78.59
DL 1	72.77	84.75	78.31
DL 2	74.86	79.87	77.28
DL 3	72.02	84.75	77.87
DL 4	72.82	86.58	79.10
DL mean	73.12	83.99	78.14
DL std	1.05	2.49	0.66

Table 1: Rule-based (RB) versus BERT-based (DL): label 1

	precision	recall	f-measure
DL 1	42.30	60.12	49.65
DL 2	51.11	58.22	54.43
DL 3	40.00	59.01	47.68
DL 4	41.11	62.71	49.66
DL mean	43.63	60.02	50.36
DL std	4.39	1.70	2.49

Table 2: Results of BERT-based (DL): label 0

mented and, thus, more is predicted. RB* gives the results if a perfect filter was applied. Out of the 183 filtered out cases⁵, 102 have triggered an inference. If we reduce the number of found cases (the denominator of precision) by 102, precision goes up to 71.83% and the f-measure raises to 78.59%. Both approaches were on par, then.

The DL model has - to a certain extent - learned that some examples belong to the category 0. Table 2 shows the DL results for the label 0. They are worse than those for 1. This might stem from the truncation which sometimes cuts away too much.

5 Related Work

Our task, detecting negative actors, is somehow related to the task of opinion role identification (see e.g. (Wiegand et al., 2019)), where the goal is to identify the source (our case) and the target of an opinion event expressed by a sentence. However, our task is more specific and more general at the same time. We are interested in opinion sources that also are conceptualized as negative actors as in *He vilifies the people*. But we not only are looking at opinion sources but are also interested in any event source (or actor) that is negatively connotated through the sentence (e.g. *He deliberately injured others*). There is a superficial similarity with the

⁵As discussed, 39.78 % of the 460 false positives could be detected by a simple filter.

work of (Wiegand et al., 2016) where also a rule-based approach was used. Our rule-based system is described in (Klenner et al., 2017). Among others, it also produces predictions of negative actors. The system uses a large verb lexicon where each verb is specified according to its various syntactic frames and where the frame elements are further specified with respect to their polar roles (e.g. negative actor) and the pro or con relation among each other. The shortcoming of the system clearly is that it is over-generating, it only partially is able to identify non-factuality and has no means to distinguish relevant from irrelevant readings of a verb.

We have also tried to find related work in the fields of stance detection and even argumentation mining. But we are not aware of any approaches that directly focuses as we do on that task, neither for German nor for any other language. The field of emotion classification is relevant, as negative actors might evoke strong emotions. Inspired by (Oberländer et al., 2020)’s paper title, we want to investigate ”which semantic roles enable machine learning to infer” the negative sentiments towards agent entities. Based on cognitive appraisal theories, the corpus of thousand sentences described in (Hofmann et al., 2020) explicit the link between emotions caused by events and the appraisal dimensions. Our ongoing attempt to quantify the severeness of negativity involved might benefit from a closer look at the emotional side. (Bostan et al., 2020) annotated emotions in English news headlines via crowdsourcing, together with semantic roles and the reader’s perception. To gain more insights into the severeness dimension, crowdsourcing could be a good way in order to come to a more representative since larger corpus.

6 Conclusion

We have introduced a dataset of 1260 actor-verb pairs (including their sentences) where each pair either identifies a negative actor of a factual situation described by the verb or the actor is not a negative actor mostly because the verb denotation - given the sentence - is non-factual. Factuality could in principle be determined on the basis of certain grammatical indicators, but other inference blockers are harder to identify (verb ambiguity). If the rule-based system had a basic (and perfect) filter, both approaches, DL and RB, are on par. We have shown that the neural models (DL) are able to learn the needed distinctions without relying on manual

feature engineering or manual filter specifications. The identification of negative actors might be useful for a system that detects offensive language or hate speech, where targets (e.g. migrants) quite often are being conceptualized as villains.

Future work will focus on the determination of the strength of negativity. Manually quantifying negativity of actors is an error prone task. Moreover, the actual strength value is not so crucial - it is the right ranking (is actor A more negative than actor B) that counts. We have started to experiment with a lexicon-based quantification metric (see (Clematide and Klenner, 2010) for the lexicon) that takes into account different types of sentiment specifications, e.g. appraisal categories (Martin and White, 2005) (judgement, emotion, apprehension words) and the classification of emotion words according to the base emotions they express (Plutchik, 1980).

7 Acknowledgements

Our work is supported by the Swiss National Foundation under the project number 105215_179302.

References

- Laura Ana Maria Bostan, Evgeny Kim, and Roman Klinger. 2020. GoodNewsEveryone: A corpus of news headlines annotated with emotions, semantic roles, and reader perception. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1554–1566, Marseille, France. European Language Resources Association.
- Simon Clematide and Manfred Klenner. 2010. Evaluation and extension of a polarity lexicon for German. In *Proceedings of the First Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA)*, pages 7–13.
- Jan Hofmann, Enrica Troiano, Kai Sassenberg, and Roman Klinger. 2020. Appraisal theories for emotion classification in text. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 125–138, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Manfred Klenner and Michael Amsler. 2016. Sentiframes: A resource for verb-centered German sentiment inference. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).
- Manfred Klenner and Simon Clematide. 2016. How factuality determines sentiment inferences. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 75–84, Berlin, Germany. Association for Computational Linguistics.
- Manfred Klenner, Don Tuggener, and Simon Clematide. 2017. Stance detection in Facebook posts of a German right-wing party. In *LSDSem 2017/LSD-Sem Linking Models of Lexical, Sentential and Discourse-level Semantics*.
- J. R. Martin and P. R. R. White. 2005. *Appraisal in English*. Palgrave, London.
- Laura Ana Maria Oberländer, Kevin Reich, and Roman Klinger. 2020. Experiencers, stimuli, or targets: Which semantic roles enable machine learning to infer the emotions? In *Proceedings of the Third Workshop on Computational Modeling of People’s Opinions, Personality, and Emotion’s in Social Media*, Barcelona, Spain (Online). Association for Computational Linguistics.
- Robert Plutchik. 1980. *A general psychoevolutionary theory of emotion*. Academic press, New York.
- Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin. 2009. A new hybrid dependency parser for German. In *Proc. of the German Society for Computational Linguistics and Language Technology*, pages 115–124.
- Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. 2009. Stylebook for the Tübingen treebank of written German. Technical report, Universität Tübingen, Seminar für Sprachwissenschaft.
- Michael Wiegand, Nadisha-Marie Aliman, Tatjana Anikina, Patrick Carroll, Margarita Chikobava, ErikHahn, Marina Haid, Katja König, Leonie Lapp, Artuur Leeuwenberg, Martin Wolf, and Maximilian Wolf. 2016. Saarland university’s participation in the second shared task on source, subjective expression and target extraction from political speeches (steps-2016). In *Proceedings of IGGSA Shared Task Workshop, Bochumer Linguistische Arbeitsberichte*.
- Michael Wiegand, Margarita Chikobava, and Josef Ruppenhofer. 2019. A supervised learning approach for the extraction of sources and targets from German text. In *Proceedings of the 15th Conference on Natural Language Processing, KONVENS 2019, Erlangen, Germany, October 9-11, 2019*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Talrómur: A large Icelandic TTS corpus

Atli Thor Sigurgeirsson

University of Edinburgh

Þorsteinn Daði Gunnarsson, Gunnar Thor Örnólfsson, Eydís Huld Magnúsdóttir
Ragnheiður Kr. Þórhallsdóttir, Stefán Gunnlaugur Jónsson, Jón Guðnason

Menntavegur 1

Reykjavík University

Abstract

We present Talrómur¹, a large high-quality Text-To-Speech (TTS) corpus for the Icelandic language. This multi-speaker corpus contains recordings from 4 male speakers and 4 female speakers of a wide range in age and speaking style. The corpus consists of 122,417 single utterance recordings equating to approximately 213 hours of voice data. All speakers read from the same script which has a high coverage of possible Icelandic diphones. Manual analysis of 15,956 utterances indicates that the corpus has a reading mistake rate no higher than 0.25%. We additionally present results from subjective evaluations of the different voices with regards to intelligibility, likeability and trustworthiness.

1 Introduction

All statistical TTS models require some training data to learn the mapping from text to speech. Unit selection TTS models are capable of producing an intelligible voice using less than 2 hours of aligned speech (Conkie, 1999). HMM-based TTS models can produce somewhat natural-sounding speech using less than 500 utterances (Yoshimura et al., 1999). The more recent neural end-to-end models have reached a considerably higher mean opinion score (MOS) in regard to naturalness. However, they require a much larger training corpus; most require tens of thousands of utterances to converge and reach natural sounding synthesis (Wang et al., 2017) (Ren et al., 2019). The widely used LJ Speech corpus consists of 13,100 recordings amounting to approximately 24 hours (Ito and Johnson, 2017).

To produce high quality synthesised speech with minimal noise, the corpora used for training TTS models are most often captured in a studio under supervision. New approaches have lowered the language-specific expertise needed for high quality TTS but at the cost of requiring larger amounts of training data (Sotelo et al., 2017) (Arik et al., 2017) (Wang et al., 2017) (Ren et al., 2019). The large amount of data needed and the quality of that data limits the ability of many low resource language communities to benefit from these recent advancements in the TTS domain.

The Icelandic language program (ILP) is a 5 year government funded program to make the Icelandic language viable in the digital world (Nikulásdóttir et al., 2020). TTS development for Icelandic is a significant part of the ILP ranging from unit selection voices to multi-speaker TTS models. A prerequisite for all TTS projects of the ILP is a large high quality TTS corpus which up to this point has not been available for open use (Nikulásdóttir et al., 2020).

Previous work in spoken language technology for Icelandic has been more focused on speech recognition, both in terms of data acquisition and acoustic modelling (Helgadóttir et al., 2017) (Guðnason et al., 2012) (Steingrímsson et al., 2017) (Mollberg et al., 2020). Since most of that data is found or crowd-sourced data from multiple speakers it is not ideal for speech synthesis where low background noise and high recording quality is important. An Icelandic pronunciation dictionary for TTS exists as well as a limited text normalisation system (Nikulásdóttir et al., 2018) (Nikulásdóttir and Guðnason, 2019). To address the lack of high quality Icelandic TTS data, Talrómur has been created.

2 The Talrómur Corpus

One of the aims of the Talrómur project is to attain diversity in age, speaking style, dialect and

¹"Tal" means speech and "rómur" means voice.

ID	Name	Gender	Age	# Utterances	Duration	# Characters	# Words	# Unique Words
A	Rósa	F	59	9,899	16h32m12s	556,767	93,002	19,272
B	Bjartur	M	70	12,048	25h43m05s	713,578	118,564	22,617
C	Diljá	F	71	13,443	27h57m33s	843,530	139,636	25,492
D	Búi	M	49	12,357	22h32m58s	766,037	126,814	23,857
E	Ugla	F	26	20,050	31h28m04s	1,298,318	215,176	33,629
F	Álfur	M	35	19,849	29h07m18s	1,284,508	212,979	33,401
G	Salka	F	33	16,886	30h09m38s	1,078,978	178,818	29,966
H	Steinn	M	39	17,637	29h49m01s	1,134,244	187,868	30,977

Table 1: Overview of corpus, outlining key statistics and information for each speaker. The "Name" column contains pseudonyms for the speakers in the corpus

prosody. Voice samples from speaker applicants were analysed and evaluated with this and a subjective evaluation of pleasantness in mind. Each participating speaker got a recording schedule, typically two hours each working day until completion.

Dialect diversity is low in Iceland and six main but rather similar regional variants are listed in the Icelandic pronunciation dictionary (Nikulásdóttir et al., 2018). Speakers A-F all speak in the most frequent standard dialect while speakers G and H speak in the second most frequent regional variant. Speakers A, B and C differ a bit from the rest of the group and their qualities deserve a specific mention.

Speaker A was the first speaker we recorded. At that time the development of the recording client was ongoing and we had limited experience with the studio and equipment. As shown in table 1 that speaker has significantly fewer hours recorded.

Speaker B is a 70 year old man with limited eyesight. This speaker often had issues with reading the prompts fluently. This results in unnatural pauses in the middle of sentences that correspond with where the line is split on the screen. We have looked into using silence detection to remove these silences and current results suggest that this task is easily automated. We release the data in the raw format however, without any trimming.

Speaker C is a female voice actor with a deep, breathy voice. This speaker’s recordings are more similar to audio-book recordings in that they have a more animated speaking style when compared to the other speakers.

Technical Details

Each speaker reads single sentence prompts from the same reading list. The reading list was de-

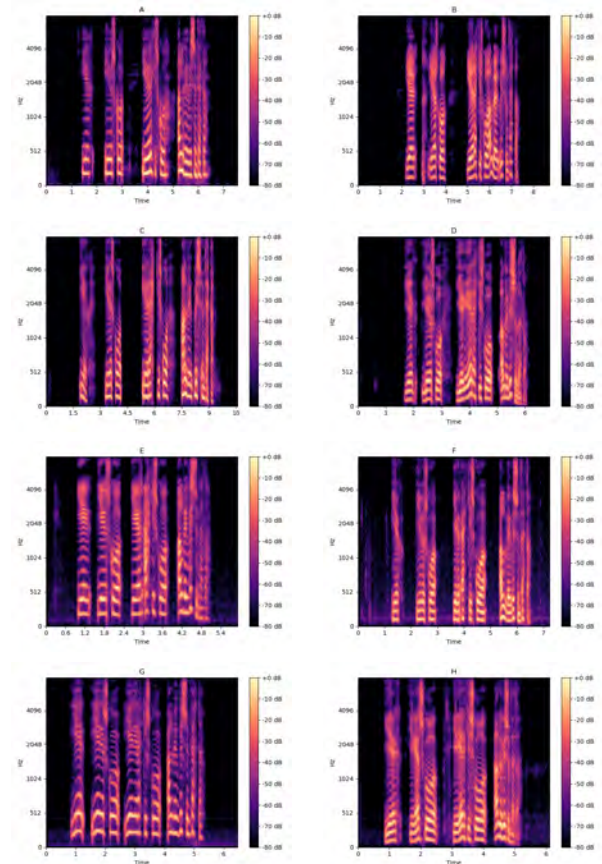


Figure 1: Mel-frequency spectrograms of all speakers saying the same phrase: "Ég, ég er sko, ég er ekki sko, alveg viss um þetta".

signed to have a high coverage of diphones in the Icelandic language (Sigurgeirsson et al., 2020). The prompts were sourced from Risamálheild, a large Icelandic text corpus consisting of text from many different types of sources (Steingrímsson et al., 2018).

Recording sessions were carried out in a stu-

dio at the national broadcaster of Iceland. After recording the first 2 speakers, the project was moved to a different studio at the national broadcaster due to restrictions caused by the COVID-19 pandemic. The last two speakers reside in northern Iceland and they were therefore recorded in a third studio. The recordings were captured between November 2019 and September 2020.

Since the speakers read prompts from the same reading list nearly all sentences in the corpus are spoken by multiple speakers. This makes the corpus ideal for multi-speaker TTS development, prosody transfer, voice conversion and other research domains where the speaker identity and linguistic content have to be disentangled by the TTS model (Skerry-Ryan et al., 2018) (Wang et al., 2018).

The same recording hardware was used for all recordings. The hardware consisted of an AKG ULS series condenser microphone equipped with a CK-61 cardioid capsule, an SPL channel one pre-amplifier and a Clarett 2Pre sound card. The recordings are captured using a recording client specifically made for this project (Sigurgeirsson et al., 2020).

We store some information about every recording captured, such as how the text appeared on the monitor to the speaker, the session ID and technical information about the recordings. Most recordings are sampled at 48kHz with a 16 bit depth. Some recordings of speakers A and B are sampled at 44.1kHz. All recordings are single channel.

3 Recording Analysis

Type of error	Occurrence	Rate
Volume too low	8	0.05%
Volume too high	70	0.44%
Audio flaw	347	2.17%
Prompt mismatch	196	1.23%
Actual mismatch	39	0.25%

Table 2: The results of 15,956 recording analyses. The evaluators judge long silences as prompt mismatches resulting in 196 prompt mismatch evaluations. Subtracting those results in a much lower number or 39.

We have analysed a portion of the recordings for quality. Of the approximately 122,417 recordings 15,956 recordings have been analysed. Using a proprietary tool, human evaluators are asked

to first listen to a single recording and then indicate whether the recording matches the prompt and whether the recording quality is good. We specifically ask the evaluators to indicate whether the volume is either too high, resulting in pops or distortions, or too low making the recording hard to comprehend or whether any other audio flaws are audible in the recording.

Of the recordings analysed 613 were marked as bad or about 3.8%. Only 1.23% of the recordings were indicated to have a mismatch between the prompt and the recording. Upon further inspection it seems that the evaluators marked recordings with untimely silences as prompt mismatches. Most of those are spoken by speaker B as explained in section 2. After a second pass over the evaluations we are confident that a better estimate of prompt mismatches is no more than 0.25%.

The rate of audio flaws is 2.17% but reviewing the samples in question revealed that a significant portion of these recordings do not have any unwanted artefacts. Upon inspection we believe some of these recordings have a higher than normal volume, making them sound unpleasant when compared to other recordings. This is particularly common for speaker B. The volume of recordings can be too high if the speaker has moved too close to the microphone, the hardware has not been configured correctly or the speaker speaks with more effort than is natural to the speaker. There are however some recordings that do have unwanted artefacts. In most cases this consists of a small pop at a random location in the recording. These pops mostly appear in recordings from speakers A and B and we therefore deduce that the source of this artefact is the hardware configuration in the recording studio for those speakers.

4 Subjective Listening Experiment

To gain further information about which voice would be most suitable for general TTS use, we set up a subjective listening experiment with 50 participants. During the listening experiment, the participants listen to a single recording at a time. They are then asked one of three questions²:

Q1: How easy is it to understand this voice?

Q2: How pleasant is this voice?

Q3: How trustworthy is this voice?

²In Icelandic: Q1: *Hversu auðskiljanleg þykir þér þessi rödd?* Q2: *Hversu viðkunnanleg þykir þér þessi rödd?* Q3: *Hversu traustverðug þykir þér þessi rödd?*

ID	SR		F0			Duration		
	words / sec	chars / sec	Min	Mean \pm SD	Max	Min	Mean \pm SD	Max
A	2.34	13.70	147.77	198.82 \pm 22.56	246.68	1.30	6.01 \pm 1.55	14.38
B	1.73	10.19	76.58	150.71 \pm 24.57	215.14	2.22	7.68 \pm 1.91	18.68
C	1.89	11.20	107.62	173.61 \pm 24.52	331.10	2.71	7.48 \pm 1.77	17.76
D	2.24	13.28	79.69	143.69 \pm 28.02	210.22	0.91	6.57 \pm 1.53	15.97
E	2.94	17.39	128.37	210.74 \pm 27.00	294.88	1.86	5.65 \pm 1.50	14.46
F	3.26	19.33	102.03	128.10 \pm 12.89	165.02	1.78	5.28 \pm 1.36	12.96
G	2.39	14.13	154.71	237.08 \pm 20.60	271.69	2.26	6.43 \pm 1.64	14.82
H	2.60	15.42	98.45	142.69 \pm 23.36	213.84	1.44	6.09 \pm 1.56	14.57

Table 3: Estimation of speaking rate (SR) and average F0. Pitch was estimated by averaging pitch over voiced segments in the phrase used in figure 1. *ProsodyPro* was used for pitch tracking (Xu, 2016).

The participants then rate the recording on a scale from 1 to 5, e.g. from very untrustworthy to very trustworthy. Before starting the evaluation participants are made aware that the sentences being spoken should not affect their judgement and that they should focus on the voice itself.

Each participant listens to 3 recordings from each speaker for each of the three questions, resulting in 24 evaluations per question and 72 evaluations in total per participant. We used a balanced Latin square experimental design with 24 different recordings tested for each evaluation question (MacKenzie, 2002). This resulted in 1074 Q1 responses, 1074 Q2 responses and 1088 Q3 responses. The number of responses per utterance ranges from 4 to 8.

Results from this experiment are shown in table 4. These scores are relative between the 8 speakers since listeners only listen to recordings from the Talrómur corpus. Due to the fact that the listening test wasn't anchored, the interpretation of the rating scale varied noticeably between listeners. The results we present here are normalised per listener, and the raw scores are higher, particularly for Q1. Voice G is rated as the most intelligible, voice H as the most likable and most trustworthy, although they didn't score significantly higher than the second highest for each question.

5 Summary and Future Work

In this paper we introduce the Talrómur corpus which is the result of the first TTS data acquisition phase of the Icelandic language program. Talrómur is a large, high quality speech corpus designed specifically for TTS. The corpus consists of 8 different voices with a wide range in prosodic effect, speaking style and age. The quality and amount of

ID	Q1	Q2	Q3
A	2.78 \pm 0.36	2.84 \pm 0.33	2.80 \pm 0.32
B	1.82 \pm 0.36	1.66 \pm 0.30	1.50 \pm 0.30
C	2.96 \pm 0.37	1.95 \pm 0.38	2.14 \pm 0.35
D	3.57 \pm 0.33	3.02 \pm 0.31	3.55 \pm 0.29
E	4.13 \pm 0.28	2.87 \pm 0.32	3.72 \pm 0.28
F	3.54 \pm 0.31	3.10 \pm 0.34	2.87 \pm 0.34
G	4.27 \pm 0.22	2.91 \pm 0.33	3.32 \pm 0.30
H	3.97 \pm 0.28	3.15 \pm 0.27	3.73 \pm 0.28

Table 4: Normalised mean opinion score with standard deviation for each speaker and each question. Q1 tested for intelligibility, Q2 for likeability and Q3 for trustworthiness.

data in Talrómur matches or exceeds that used in many state-of-the-art end-to-end neural TTS models for the English language. A subjective evaluation indicates which voice users are likely to prefer but we believe most of the voices are good candidates for general TTS use. As with other deliverables belonging to the ILP, the data will be published under open licenses to encourage wide use and adoption of the data. The data has been made available through the CLARIN project³.

6 Acknowledgements

This project was funded by the Language Technology Programme for Icelandic 2019-2023. The programme, which is managed and coordinated by Almannarómur⁴, is funded by the Icelandic Ministry of Education, Science and Culture.

³<https://repository.clarin.is/repository/xmlui/handle/20.500.12537/104>

⁴<https://almannaromur.is/>

References

- Sercan O Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. 2017. Deep voice: Real-time neural text-to-speech.
- Alistair Conkie. 1999. Robust unit selection system for speech synthesis. In *137th meeting of the Acoustical Society of America*, page 978.
- Jón Guðnason, Oddur Kjartansson, Jökull Jóhannsson, Elín Carstensdóttir, Hannes Högni Vilhjálmsson, Hrafn Loftsson, Sigrún Helgadóttir, Kristín M Jóhannsdóttir, and Eiríkur Rögnvaldsson. 2012. Almennarmur: An open Icelandic speech corpus. In *Spoken Language Technologies for Under-Resourced Languages*.
- Inga Rún Helgadóttir, Róbert Kjaran, Anna Björk Nikulásdóttir, and Jón Guðnason. 2017. Building an asr corpus using althingi’s parliamentary speeches. In *INTERSPEECH*, pages 2163–2167.
- Keith Ito and Linda Johnson. 2017. The lj speech dataset. <https://keithito.com/LJ-Speech-Dataset/>.
- I Scott MacKenzie. 2002. Within-subjects vs. between-subjects designs: Which to use? *Human-Computer Interaction: An Empirical Research Perspective*, 7:2005.
- David Erik Mollberg, Ólafur Helgi Jónsson, Sunneva Þorsteinsdóttir, Steinþór Steingrímsson, Eydís Huld Magnúsdóttir, and Jon Gudnason. 2020. Samrómur: Crowd-sourcing data collection for Icelandic speech recognition. In *Proceedings of the 12th Conference on Language Resources and Evaluation*, pages 3463–3467.
- Anna Björk Nikulásdóttir and Jón Guðnason. 2019. Bootstrapping a Text Normalization System for an Inflected Language. Numbers as a Test Case. In *INTERSPEECH*, pages 4455–4459.
- Anna Björk Nikulásdóttir, Jón Guðnason, Anton Karl Ingason, Hrafn Loftsson, Eiríkur Rögnvaldsson, Einar Freyr Sigurðsson, and Steinþór Steingrímsson. 2020. Language technology programme for Icelandic 2019–2023. page 3414–3422.
- Anna Björk Nikulásdóttir, Jón Guðnason, and Eiríkur Rögnvaldsson. 2018. An Icelandic pronunciation dictionary for tts. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 339–345. IEEE.
- Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech. In *Advances in Neural Information Processing Systems*, pages 3171–3180.
- Atli Sigurgeirsson, Gunnar Örnólfsson, and Jón Guðnason. 2020. Manual speech synthesis data acquisition-from script design to recording speech. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 316–320.
- RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous. 2018. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *international conference on machine learning*, pages 4693–4702. PMLR.
- Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. 2017. Char2wav: End-to-end speech synthesis. In *In ICLR2017 workshop submission*.
- Steinþór Steingrímsson, Jón Guðnason, Sigrún Helgadóttir, and Eiríkur Rögnvaldsson. 2017. Málrómur: A manually verified corpus of recorded Icelandic speech. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 237–240.
- Steinþór Steingrímsson, Sigrún Helgadóttir, Eiríkur Rögnvaldsson, Starkaður Barkarson, and Jón Guðnason. 2018. Risamálheild: A very large Icelandic text corpus. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*.
- Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017. Tacotron: Towards end-to-end speech synthesis. pages 4006–4010.
- Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A Saurous. 2018. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International Conference on Machine Learning*, pages 5180–5189. PMLR.
- Y Xu. 2016. Prosodypro. a praat script for large-scale systematic analysis of continuous prosodic events. In *In Proceedings of Tools and Resources for the Analysis of Speech Prosody (TRASP 2013)*.
- Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. 1999. Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. In *Sixth European Conference on Speech Communication and Technology*.

NorDial: A Preliminary Corpus of Written Norwegian Dialect Use

Jeremy Barnes*

Department of Informatics
University of Oslo
jeremycb@uio.no

Petter Mæhlum*

Department of Informatics
University of Oslo
petterma@uio.no

Samia Touileb*

Department of Informatics
University of Oslo
samiat@uio.no

Abstract

Norway has a large amount of dialectal variation, as well as a general tolerance to its use in the public sphere. There are, however, few available resources to study this variation and its change over time and in more informal areas, *e.g.* on social media. In this paper, we propose a first step to creating a corpus of dialectal variation of written Norwegian. We collect a small corpus of tweets and manually annotate them as Bokmål, Nynorsk, any dialect, or a mix. We further perform preliminary experiments with state-of-the-art models, as well as an analysis of the data to expand this corpus in the future. Finally, we make the annotations and models available for future work.

1 Introduction

Norway has a large tolerance towards dialectal variation (Bull et al., 2018) and, as such, one can find examples of dialectal use in many areas of the public sphere, including politics, news media, and social media. Although there has been much variation in writing Norwegian, since the debut of Nynorsk in the 1850's, the acceptance of dialect use in certain settings is relatively new. The official language policy after World War 2 was to include forms belonging to all layers of society into the written norms, and a “dialect wave” has been going on since the 1970's (Bull et al., 2018, 235-238).

From 1980 to 1983 there was an ongoing project called *Den første lese- og skriveopplæring på dialekt* ‘The first training in reading and writing in dialect’ (Bull, 1985), where primary school students were allowed to use their own dialect in school, with Tove Bull as project leader. Bull et al.

(2018) also point out that later interest in writing in dialect in media such as e-mail and text messages can be seen as an extension of the interest in dialectal writing in the 1980s (Bull et al., 2018, 239). They also note that the tendency has been the strongest in the county of Trøndelag initially, but later spreading to other parts of the country, also spreading among adults.

At the same time, there are two official main writing systems, *i.e.* Bokmål and Nynorsk, which offer prescriptive rules for how to write the spoken variants. This leads to a situation where people who typically use their dialect when speaking often revert to one of the written standards when writing. However, despite there being only two official writing systems, there is considerable variation within each system, as the result of years of language policies. Today we can find both ‘radical’ and ‘conservative’ versions of each writing system, where the radical ones try to bridge the gap between the two norms, while the conservative versions attempt to preserve differences. However, it is still natural that these standards have a regularizing effect on the written varieties of people who normally speak their dialect in most situations (Gal, 2017). As such, it would be interesting to know *to what degree dialect users deviate from these established norms and use dialect traits when writing informal texts, e.g.* on social media. This could also provide evidence of the vitality of certain dialectal traits.

In this paper, we propose a first step towards creating a corpus of written dialectal Norwegian by identifying the best methods to collect, clean, and annotate tweets into Bokmål, Nynorsk, or dialectal Norwegian. We concentrate on geolects, rather than sociolects, as we observe these are easier to collect on Twitter, *i.e.* the traits that identify a geolect are more likely to be written than those that identify a sociolect. This is a necessary simplification, as dialect users rarely write with

*The authors have equal contribution.

full phonetic awareness, making it impossible to find dialect traits that lie mainly in the phonology. As such, our corpus relies more on lexical and clear phonetic traits to determine whether a tweet is written in a dialect.

We collect a corpus of 1,073 tweets which are manually annotated as *Bokmål*, *Nynorsk*, *Dialect*, or *Mixed* and perform a first set of experiments to classify tweets as containing dialectal traits using state-of-the-art methods. We find that fine-tuning a Norwegian BERT model (NB-BERT) leads to the best results. We perform an analysis of the data to find useful features for searching for tweets in the future, confirming several linguistic observations of common dialectal traits and find that certain dialectal traits (those from Trøndelag) are more likely to be written, suggesting that since their traits strongly diverge from *Bokmål* and *Nynorsk*, they are more likely to deviate from the established norms when composing tweets. Finally, we release the annotations and dialect prediction models for future research.¹

2 Related Work

The importance of incorporating language variation into natural language processing approaches has gained visibility in recent years. The VarDial workshop series deals with computational methods and language resources for closely related languages, language varieties, and dialects and have offered shared tasks on language variety identification for Romanian, German, Uralic languages (Zampieri et al., 2019), among others. Similarly, there have been shared tasks on Arabic dialect identification (Bouamor et al., 2019; Abdul-Mageed et al., 2020). To our knowledge, however, there are no available written dialect identification corpora for Norwegian.

Many successful approaches to dialect identification use linear models (*e.g.* Support Vector Machines, Multinomial Naive Bayes) with word and character *n*-gram features (Wu et al., 2019; Jauhainen et al., 2019a), while neural approaches often perform poorly (Zampieri et al., 2019) (see Jauhainen et al. (2019b) for a full discussion). More recent uses of pretrained language models based on transformer architectures (Devlin et al., 2019), however, have shown promise (Bernier-Colborne et al., 2019).

¹Available at https://github.com/jerbarnes/norwegian_dialect

Corpus-related work on Norwegian dialects has mainly focused on spoken varieties. There are two larger corpora available for Norwegian: the newer Nordic Dialect Corpus (Johannessen et al., 2009), which contains spoken data from several Nordic languages, and the Language Infrastructure made Accessible (LIA) Corpus, which in addition to Norwegian also contains Sámi language clips.² There is also the *Talk of Norway* Corpus (Lapponi et al., 2018), which contains transcriptions of parliamentary speeches in many language varieties. While they contain rich dialectal information, this information is not kept in writing, as they are normalized to *Bokmål* and *Nynorsk*. These resources are useful for working with speech technology and questions about Norwegian dialects as they are spoken, but they are likely not sufficient to answer research questions about how dialects are expressed when written. The transcriptions in these corpora also differ from written dialect sources in the sense that they are in a way truer representations of the dialects in question. In writing dialect representations tend to focus more on a few core words, even if the actual phonetic realization of certain words could have been marked in writing.

3 Data collection

In this first round of annotations, we search for tweets containing *Bokmål*, *Nynorsk*, and *Dialect* terms (See Appendix A), discarding tweets that are shorter than 10 tokens. The terms were collected by gathering frequency bigram lists from the Nordic Dialect Corpus (Johannessen et al., 2009) from the written representation of the dialectal varieties.

Two native speakers annotated these tweets with four labels: *Bokmål*, *Nynorsk*, *Dialect*, and *Mixed*. The *Mixed* class refers to tweets where there is a clear separation of dialectal and non-dialectal texts, *e.g.* reported speech in *Bokmål* with comments in *Dialect*. This class can be very problematic for our classification task, as the content can be a mix of all the other three classes. We nevertheless keep it, as it still reflects one of the written representations of Norwegian.

In Example 1, we show two phrases from the Nordic Dialect Corpus, from a speaker in Ballangen, Nordland county. We show it in dialectal

²<https://www.hf.uio.no/iln/english/research/projects/language-infrastructure-made-accessible/>

	Bokmål	Nynorsk	Dialect	Mixed	Total
Train	348	174	274	52	848
Dev	52	20	30	4	106
Test	38	31	35	6	110
Total	438	225	348	62	1,073

Table 1: Data statistics for the corpus, including number of tweets per split.

form (a) and the Bokmål (b) transcription, but with added punctuation marks. To exemplify the two other categories we have manually translated it to Nynorsk (c) and added a mixed version (d), as well as an English translation (e) for reader comprehension.

- (1) (a) Æ ha løsst å fær dit. Æ har løsst å gå på skole dær.
- (b) Jeg har lyst å fare dit. Jeg har lyst å gå på skole der.
- (c) Eg har lyst å fara dit. Eg har lyst å gå på skule der.
- (d) Æ ha løsst å fær dit. Jeg har lyst å gå på skole der.
- (e) I want to go there. I want to go to school there.

The two annotators doubly annotated a subset of the data in order to assess inter annotator agreement. On a subset of 126 tweets, they achieved a Cohen’s Kappa score of 0.76, which corresponds to substantial agreement. Given the strong agreement on this subset, we did not require double annotations for the remaining tweets. Table 1 shows the final distribution of tweets in the training, development, and test splits. Bokmål tweets are the most common, followed by Dialect and Nynorsk, and as can be seen, Mixed represents a smaller subset of the data.

Certain traits made the annotation difficult. Many tweets, especially those written in dialect, are informal, and therefore contain more slang and spelling mistakes. For example, *jeg* ‘I’ can be misspelled as *eg*, which if found in a non-Nynorsk setting could indicate dialectal variation. Spelling mistakes should not interfere with dialect identification, but as some tweets can contain as little as one token that serve to identify the language variety as dialectal, this can cause problems. Some dialects are also quite similar to either Bokmål or

Bokmål-Dialect		Nynorsk-Dialect	
‘e’	288.7	‘e’	131.8
‘æ’	188.0	‘æ’	92.5
‘ska’	55.0	‘ska’	23.9
‘hu’	36.6	‘ei’	18.9
‘te’	28.9	‘berre’	14.5
(‘æ’, ‘e’)	27.5	‘hu’	14.4
‘ka’	22.0	‘heilt’	13.8
‘mæ’	21.6	(‘æ’, ‘e’)	13.2
‘går’	19.9	‘meir’	12.3
‘va’	12.4	‘mæ’	11.9

Table 2: Top 10 features and χ^2 values between Bokmål – Dialect tweets and Nynorsk – Dialect.

Nynorsk, and speakers might switch between them when speaking or writing. Similarly, certain elements can be indicative of either a geolect or a sociolect, *e.g.* the pronoun *dem* ‘they’ as the third person plural subject pronoun (*de* in Bokmål and Nynorsk), which in a rural setting might be typical for an East Norwegian dialect, while in an urban setting might be a strong sociolectal indicator. Tweets with similar problems are annotated in favor of the dialect class. Additionally, there is the problem of internal variation. A tweet can belong to a radical or conservative variety of standardized Norwegian, *e.g.* Riksmål, and thereby not be dialectal. However, this distinction can be difficult to make if a writer uses forms that are now removed from the main standards (Bokmål and Nynorsk), and therefore become more marked, such as *sprog* instead of *språk* ‘language’.

4 Dialectal traits

To find the most salient written dialect traits compared to Bokmål and Nynorsk, we perform a χ^2 test (Pearson, 1900) on the occurrence of unigrams, bigrams, and trigrams pairwise between Bokmål and Dialect, and then Nynorsk and Dialect and set $p = 0.5$.

The most salient features (see Table 2) are mainly unigrams that contain dialect features, *e.g.* *æ* ‘I’, *e* ‘am/is/are’, *ska* ‘shall/will’, *te* ‘to’, *mæ* ‘me’, *frå* ‘from’, although there are also two statistically significant bigrams, *e.g.* *æ e* ‘I am’, *æ ska* ‘I will’. We notice that many of these features likely correspond to Trøndersk and Nord-

norsk variants. Similar features from other dialects (*i, jæ, je* ‘I’) are not currently found in the corpus. This may reflect the natural usage, but it is also possible that the original search query should be improved. Example 2 shows an example of a Dialect tweet (the English translation is ‘Now you know how I’ve felt for a few years’) where the **dialectal words** have been highlighted in red and **marked words**, which are not necessarily dialectal, but which often help with classification, have been highlighted in green.

(2) Nå vet du **åssen** **æ** har hatt
det i noen år 😞

5 Experiments

We propose baseline experiments on a 80/10/10 split for training, development and testing and use a Multinomial Naive Bayes (MNB) and a linear SVM. As features, we use tf-idf word and character (1-5) n-gram features, with a minimum document frequency of 5 for words, and 2 for characters. We use MNB with $\alpha=0.01$, and SVM with hinge loss and regularization of 0.5 and use grid search to identify the best combination of parameters and features.

We also compare two Norwegian BERT models: NorBERT³ (Kutuzov et al., 2021) and NB-BERT⁴ (Kummervold et al., 2021), which use the same architecture as BERT base cased (Devlin et al., 2019). NorBERT uses a 28,600 entry Norwegian-specific sentence piece vocabulary and was jointly trained on 200M sentences in Bokmål and Nynorsk, while NB-BERT uses the vocabulary from multilingual BERT and is trained on 18 billion tokens from a variety of sources⁵, including historical texts, which presumably contain more examples of written dialect. We use the huggingface transformers implementation and feed the final ‘[CLS]’ embedding to a linear layer, followed by a softmax for classification. The only hyperparameter we optimize is the number of training epochs. We use weight decay on all parameters except for the bias and layer norms and set the learning rate for AdamW (Loshchilov and Hutter, 2019) to $1e-5$ and set all other hyperparameters to default settings. We train the model for 20 epochs,

³<https://huggingface.co/lagoslo/norbert>

⁴<https://huggingface.co/NbAiLab/nb-bert-base>

⁵See <https://github.com/NbAiLab/notram>.

		Precision	Recall	F ₁
DEV	MNB	0.70	0.67	0.68
	SVM	0.87	0.69	0.73
	NorBERT	0.73	0.72	0.72
	NB-BERT	0.89	0.90	0.89
TEST	MNB	0.60	0.61	0.60
	SVM	0.86	0.67	0.69
	NorBERT	0.73	0.72	0.72
	NB-BERT	0.81	0.78	0.79

Table 3: Precision, recall, and macro F₁ for each model, on the dev and test sets.

BK	36	0	0	2
NN	0	28	3	0
DI	2	1	32	0
MIX	0	0	4	2
	BK	NN	DI	MIX

Figure 1: Confusion matrix of NB-BERT on Bokmål (BK), Nynorsk (NN), Dialect (DI), and Mixed (MIX).

and keep the model that achieves the best macro F₁ on the dev set.

Table 3 shows the results for all models. MNB is the weakest model on both dev and test on all metrics. Despite the fact that it usually gives good results for dialect identification, it is quite clear that it does not fit our dataset. We think that this might mainly be due to the large vocabulary overlap between the classes, especially in the Mixed class. SVM has the best precision on test (0.86), while recall is lower (0.67). NB-BERT has the best recall on both dev and test (0.90/0.78), best precision on dev (0.89), and is the best overall model on test F₁ (0.79), followed by NorBERT.

6 Error analysis

Figure 1 shows a confusion matrix of NB-BERT’s predictions on the test data. The main three categories (Bokmål, Nynorsk, and Dialect) are

generally well predicted, while `Mixed` is currently the hardest category to predict. This is expected, as the `Mixed` class comprises all of the three other forms. The model has a tendency to predict `Nynorsk` or `Mixed` for `Dialect` and struggles with `Mixed`, predicting either `Bokmål` or `Dialect`. The same observations apply to `NorBERT`, `MNB`, and `SVM` classifiers.

Given that our main interest lies in the ability to predict future `Dialect` tweets, we compute precision, recall, and F_1 on only this label. The `NB-BERT` model achieves 0.82, 0.91, and 0.86, respectively while `NorBERT` follows with 0.84, 0.77, and 0.81. The `SVM` model achieves 0.80, 0.69, and 0.74 respectively, while `MNB` obtains slightly less scores with respectively 0.77, 0.66, and 0.71. This suggests that future experiments should consider using `NB-BERT`.

7 Conclusion and Future Work

In this paper we have described our first annotation effort to create a corpus of dialectal variation in written Norwegian. In the future, we plan to use our trained models to expand the corpus in a semi-supervised fashion by refining our searches for tweets with dialectal traits in order to have a larger corpus of dialectal tweets, effectively pursuing a high-precision low-recall path. In parallel, we will begin to download large numbers of tweets and use our trained models to automatically annotate these (low-precision, high-recall). At the same time we plan to perform continuous manual evaluations of small amounts of the data in order to identify a larger variety of dialectal tweets, which we will incorporate into the training data for future models.

Second, we would like to annotate these dialectal tweets with their specific dialect. To avoid collecting too many tweets from overrepresented dialects, we will first annotate the current dialectal tweets with their dialect, and perform a balanced search to find a similar number of tweets for each dialect.

Finally, we would like to incorporate texts from different sources which contain rich dialectal variation, as *e.g.* books, music, poetry.

References

- Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020. NADI 2020: The first nuanced Arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 97–110, Barcelona, Spain (Online). Association for Computational Linguistics.
- Gabriel Bernier-Colborne, Cyril Goutte, and Serge Léger. 2019. Improving cuneiform language identification with BERT. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 17–25, Ann Arbor, Michigan. Association for Computational Linguistics.
- Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The MADAR shared task on Arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207, Florence, Italy. Association for Computational Linguistics.
- Tove Bull. 1985. *Lesing og barns talemål*. Novus, Oslo.
- Tove Bull, Espen Karlsen, Eli Raanes, and Rolf Theil. 2018. *Norsk språkhistorie*, volume 3. Novus, Oslo.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Susan Gal. 2017. Visions and revisions of minority languages: Standardization and its dilemmas. In Pia Lane, James Costa, and Haley de Korne, editors, *Standardizing Minority Languages: Competing Ideologies of Authority and Authenticity in the Global Periphery*, pages 222–242. Routledge.
- Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2019a. Discriminating between Mandarin Chinese and Swiss-German varieties using adaptive language models. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 178–187, Ann Arbor, Michigan. Association for Computational Linguistics.
- Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2019b. Language model adaptation for language and dialect identification of text. *Natural Language Engineering*, 25(5):561–583.
- Janne Bondi Johannessen, Joel James Priestley, Kristin Hagen, Tor Anders Åfarli, and Øystein Alexander Vangnes. 2009. The nordic dialect corpus— an advanced research tool. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, pages 73–80, Odense, Denmark. Northern European Association for Language Technology (NEALT).

- Per Egil Kummervold, Javier de la Rosa, Freddy Wetjen, and Svein Arne Brygfeld. 2021. Operationalizing a national digital library: The case for a norwegian transformer model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa 2021)*.
- Andrey Kutuzov, Jeremy Barnes, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2021. Large-scale contextualised language modelling for norwegian. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa 2021)*.
- Emanuele Lapponi, Martin Søyland, Erik Velldal, and Stephan Oepen. 2018. The talk of norway: a richly annotated corpus of the norwegian parliament, 1998–2016. *Language Resources and Evaluation*, 52.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Karl Pearson. 1900. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175.
- Nianheng Wu, Eric DeMattos, Kwok Him So, Pin-zhen Chen, and Çağrı Çöltekin. 2019. Language discrimination and transfer learning for similar languages: Experiments with feature combinations and adaptation. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 54–63, Ann Arbor, Michigan. Association for Computational Linguistics.
- Marcos Zampieri, Shervin Malmasi, Yves Scherrer, Tanja Samardžić, Francis Tyers, Miikka Silfverberg, Natalia Klyueva, Tung-Le Pan, Chu-Ren Huang, Radu Tudor Ionescu, Andrei M. Butnaru, and Tommi Jauhiainen. 2019. A report on the third VarDial evaluation campaign. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–16, Ann Arbor, Michigan. Association for Computational Linguistics.

A Appendix

‘jæi går’, ‘e gå’.

Bokmål terms: ‘jeg har’, ‘de går’, ‘jeg skal’, ‘jeg blir’, ‘de skal’, ‘jeg er’, ‘de blir’, ‘de har’, ‘de er’, ‘dere går’, ‘dere skal’, ‘dere blir’, ‘dere har’, ‘dere er’, ‘hun går’, ‘hun skal’, ‘hun blir’, ‘hun har’, ‘hun er’, ‘jeg går’.

Nynorsk terms: ‘eg har’, ‘dei går’, ‘eg skal’, ‘eg blir’, ‘dei skal’, ‘eg er’, ‘dei blir’, ‘dei har’, ‘dei er’, ‘de går’, ‘dykk går’, ‘de skal’, ‘dykk skal’, ‘de blir’, ‘dykk blir’, ‘de har’, ‘dykk har’, ‘de er’, ‘dykk er’, ‘ho gaar’, ‘ho skal’, ‘ho blir’, ‘ho har’, ‘ho er’, ‘eg går’.

Dialect terms: ‘e ha’, ‘æ ha’, ‘æ har’, ‘e har’, ‘jæ ha’, ‘eg har’, ‘eg ha’, ‘je ha’, ‘jæ har’, ‘di går’, ‘demm går’, ‘dem går’, ‘dæmm går’, ‘dæm går’, ‘dæi går’, ‘demm gå’, ‘dem gå’, ‘di går’, ‘domm gå’, ‘dom gå’, ‘dømm går’, ‘døm går’, ‘dæmm gå’, ‘dæm gå’, ‘e ska’, ‘æ ska’, ‘jæ ska’, ‘eg ska’, ‘je ska’, ‘i ska’, ‘ei ska’, ‘jæi ska’, ‘je skæ’, ‘e bli’, ‘æ bli’, ‘jæ bli’, ‘e bi’, ‘æ blir’, ‘æ bi’, ‘je bli’, ‘e blir’, ‘i bli’, ‘di ska’, ‘dæmm ska’, ‘dæm ska’, ‘dæi ska’, ‘demm ska’, ‘dem ska’, ‘domm ska’, ‘dom ska’, ‘dømm ska’, ‘døm ska’, ‘dæ ska’, ‘domm ska’, ‘dom ska’, ‘æmm ska’, ‘æm ska’, ‘eg e’, ‘æ e’, ‘e e’, ‘jæ æ’, ‘e æ’, ‘jæ ær’, ‘je æ’, ‘i e’, ‘æg e’, ‘di bi’, ‘di bli’, ‘dæi bli’, ‘dæmm bli’, ‘dæm bli’, ‘di blir’, ‘demm bli’, ‘dem bli’, ‘dæmm bi’, ‘dæm bi’, ‘dømm bli’, ‘døm bli’, ‘dømm bi’, ‘døm bi’, ‘di har’, ‘di ha’, ‘dæmm ha’, ‘dæm ha’, ‘dæmm har’, ‘dæm har’, ‘dæi he’, ‘demm har’, ‘dem har’, ‘demm ha’, ‘dem ha’, ‘dæi ha’, ‘di he’, ‘dæmm e’, ‘dæm e’, ‘di e’, ‘dæi e’, ‘demm e’, ‘dem e’, ‘di æ’, ‘dømm æ’, ‘døm æ’, ‘demm æ’, ‘dem æ’, ‘dei e’, ‘dæi æ’, ‘dåkk går’, ‘dåkke går’, ‘dåkke gå’, ‘de går’, ‘dåkk ska’, ‘dere ska’, ‘dåkker ska’, ‘dåkke ska’, ‘di ska’, ‘de ska’, ‘åkk ska’, ‘røkk ska’, ‘døkker ska’, ‘døkk bli’, ‘dåkker bi’, ‘dåkke bli’, ‘dåkker har’, ‘dåkker ha’, ‘dere ha’, ‘dåkk ha’, ‘de har’, ‘dåkk har’, ‘dere har’, ‘de ha’, ‘døkk ha’, ‘dåkker e’, ‘dåkk e’, ‘dåkke e’, ‘di e’, ‘dere ær’, ‘dåkk æ’, ‘de e’, ‘økk e’, ‘døkk æ’, ‘ho går’, ‘hu går’, ‘ho jennng’, ‘ho gjennng’, ‘u går’, ‘o går’, ‘ho jænng’, ‘ho gjænng’, ‘ho jennng’, ‘ho gjennng’, ‘ho jenne’, ‘ho gjenne’, ‘ho gå’, ‘ho ska’, ‘hu ska’, ‘a ska’, ‘u ska’, ‘o ska’, ‘hu skar’, ‘honn ska’, ‘ho sjka’, ‘hænne ska’, ‘ho bli’, ‘ho bi’, ‘o bli’, ‘ho blir’, ‘hu bli’, ‘hu bler’, ‘hu bi’, ‘ho bir’, ‘a blir’, ‘ho ha’, ‘ho har’, ‘ho he’, ‘hu har’, ‘hu ha’, ‘hu he’, ‘o har’, ‘o ha’, ‘hu e’, ‘ho e’, ‘hu e’, ‘ho æ’, ‘hu æ’, ‘o e’, ‘hu ær’, ‘u e’, ‘ho ær’, ‘ho er’, ‘e går’, ‘æ går’, ‘eg går’, ‘jæ gå’, ‘jæ går’, ‘æ gå’,

The Swedish Winogender Dataset

Saga Hansson¹ Konstantinos Mavromatakis¹
Yvonne Adesam² Gerlof Bouma² Dana Dannélls²

¹ University of Gothenburg

² Språkbanken Text, Department of Swedish, University of Gothenburg

{sagawhansson, konstantinosmavromatakis}@gmail.com

{yvonne.adesam, gerlof.bouma, dana.dannells}@svenska.gu.se

Abstract

We introduce the SweWinogender test set, a diagnostic dataset to measure gender bias in coreference resolution. It is modelled after the English Winogender benchmark, and is released with reference statistics on the distribution of men and women between occupations and the association between gender and occupation in modern corpus material. The paper discusses the design and creation of the dataset, and presents a small investigation of the supplementary statistics.

1 Introduction

Winogender (Rudinger et al., 2018) is a diagnostic dataset designed to detect gender bias in English language coreference resolution systems, inspired by the Winograd Schema Challenge (Levesque et al., 2012). It is also found as part of SuperGlue, a set of benchmark tasks for evaluating Natural Language Understanding models (Wang et al., 2019).¹ Unlike Winograd-style test sets, Winogender is not meant to be a particularly challenging pronoun resolution test set per se, but to lay bare a specific type of gender bias in systems.

Sentences in the Winogender test set contain pronouns whose interpretation is fully determined by causal reasoning. Each sentence contains two noun phrases that could, as far as syntax is concerned, serve as antecedents for the pronoun, one introducing a referent by their occupation, and the other a further participant, which alternatively is referred to with an indefinite pronoun. Furthermore, the sentences are given in several variants, with pronouns with different gender agreement properties (*he*, *she*, [singular] *they*). Examples 1 and 2 are illustrative of the type of sentences in the Winogender test set. Coreferents are in bold.

¹<https://super.gluebenchmark.com/>

- (1) **The paramedic** performed CPR on *the passenger/someone* even though **she/he/they** knew it was too late.
- (2) *The paramedic* performed CPR on **the passenger/someone** even though **she/he/they** was/were already dead.

A crucial aspect of the Winogender sentences is that their interpretation does not depend on the form of the pronoun. So, from common sense reasoning alone – and the assumption that no further entities are relevant – one can conclude that the three alternative pronouns in (1) should all refer to the paramedic, whereas the three alternative pronouns in (2) refer to the other participant (that is, the passenger/someone). In particular, the extent to which the mentioned occupation is perceived as associated with men or women does *not* influence the interpretation of the pronoun.

By inspecting the performance of a pronoun resolution system on the different sentence variants, we can assess the gender-occupation bias inherent in the system. For an unbiased system, there should not be a difference in performance between the pronominal forms. In addition, Rudinger et al. (2018) look at the correlation of model prediction with measures of the binary gender association of the occupations in the test set. For three pronoun resolution systems, the comparisons show clear over-tendencies to resolve the pronoun *she* to female-associated occupations, and under-tendencies to resolve *she* to male-associated occupations.

In this paper, we introduce *SweWinogender*, a Swedish pronoun resolution test set modelled on the Winogender resource. The test set includes Swedish sentences of the type exemplified above. In addition, we provide occupation-gender association statistics relevant to the Swedish language and the Swedish society. Following Rudinger et al. (2018), we supply real-world statistics as well as

corpus-based statistics. The dataset is made available under an open license.²

For English, several other studies and benchmarks consider gender-bias in pronoun resolution systems. Zhao et al. (2018, WinoBias) and Lu et al. (2020) use constructed, templatic test items like Winogender, and also investigate ways to mitigate the observed biases. The latter paper presents a slightly different methodology, as bias is not assessed through model predictions, but by looking at model scores. Webster et al. (2018) and Cao and Daumé III (2020) present curated test sets compiled from attested material, with items that lack distinguishing gender-related cues. In addition, the latter moves beyond a binary perspective on gender, and includes a discussion of the harm gender biases in pronoun resolution systems may cause. Beyond English, however, not much directly related work exists. Stanovsky et al. (2019) use the English Winogender and WinoBias sets to probe gender bias in machine translation systems. We are unaware of any previous work that specifically targets gender-bias in coreference resolution systems for languages other than English.

The rest of this paper is structured as follows. We start by presenting the approach taken to create the resource (Section 2). We then describe our real-world occupational gender statistics (Section 3) for Sweden. We continue by exploring gender in the Swedish Culturomics Gigaword corpus (Section 4) and end with conclusion and pointers to future work.

2 Creating SweWinogender

The English Winogender sentences were formulated with the intent that changing the gender of a pronoun should not affect its resolution. The causal/logical structures of the sentences are carefully crafted such that pronoun interpretation is as unambiguous as possible for humans. A Mechanical Turk experiment confirmed that the sentences were indeed unambiguous (Rudinger et al., 2018). To avoid having to reinvent scenarios that have this property, we modelled the SweWinogender collection on the English original.

The English templates were loosely translated into Swedish templates, which then each give rise to twelve similar Swedish sentences: two continuations that force different readings \times two ways

²<https://spraakbanken.gu.se/en/resources/swewinogender>

of referring to the participant (using a descriptive noun or using *någon* ‘someone’) \times three pronouns (*han* ‘he’, *hon* ‘she’, *hen* ‘(singular) they’ – or object/possessive forms where appropriate). The Swedish dataset contains 624 sentences in total. Examples 3 and 4 below are taken from the Swedish Winogender dataset. The two sentences each contain three mentions: the occupation *läkaren* ‘the physician’, the participant *patienten* ‘the patient’, and the pronoun *hen*. In the first example the pronoun corefers with the participant, in the second with the occupation. Each such sentence occurs six times, three with the specific participant and each of the three pronouns to be resolved, and three with the generic participant *någon* ‘someone’ and each of the three pronouns to be resolved.

- (3) *Läkaren* sa till **patienten** att **hen**
The physician told the patient that they
behövde mer vila.
needed more rest.
- (4) **Läkaren** sa till *patienten* att **hen**
The physician told the patient that they
inte kunde skriva ut en högre läkemedelsdos.
could not prescribe a higher dose of medicine.

Sometimes the English occupation was not easily translated to Swedish, because of differences between the American and Swedish contexts. Since our goal was not to create an exact translation, we chose other roles to fit the logic in the discourses. In a number of cases we had to reformulate Swedish sentences due to linguistic differences between Swedish and English. A problematic class of sentences contained possessive pronouns, that potentially corefered with the closest subject. In Swedish, subject coreferring possessives are reflexive possessives, and these are unmarked for gender of the referent, which makes them unsuitable as a diagnostic for gender bias. A second problem with possessives is that regular possessives alternate with reflexive possessives depending on whether there is coreference with the nearest subject or not. This means that even regular possessives may be syntactically unambiguous, making them unsuitable for a diagnostic that relies on syntactic – but not pragmatic – ambiguity. This alternation is illustrated in the following sentence:

- (5) X träffade Y för att diskutera sina_X /
X met Y to discuss POSS-REFL
hans_Y/hennes_Y/hens_Y framsteg
his/her/their progress

Finally, there is the issue of the inclusion of a gender-neutral neutral pronoun in the test items. English has a relatively well-established gender-neutral pronoun in the form of (singular) *they/them/their*. For Swedish, there has been quite a lot of public debate in the last decade or so about the gender-neutral *hen/hens*. It is not common to introduce new pronouns in a language, but *hen* appears to have weathered out objections. Since 2015 it is even included in the glossary published by the Swedish Academy (SAOL). Unlike *they*, *hen* is unambiguously singular. We have used it for SweWinogender, but considering its rise in use is only recent, it may not be as useful for systems based on older texts.

3 Real-world statistics on gender and occupation

An important part of the diagnostic potential of the Winogender test set is the availability of statistics on the distribution of gender across occupations. It allows a more fine-grained investigation of the correlation of system behaviour with gender biases, by seeing if system predictions follow the distribution of genders for the occupation in a test item. Statistics on gender and occupation also highlight a subset of the Winograd sentences as particularly worthy of close scrutiny, namely those for which the gender bias strongly goes against the intended interpretation of the pronoun. We refer the reader to the original Winogender and WinoBias papers for worked-out examples of the diagnostic methodology (Rudinger et al., 2018; Zhao et al., 2018). The methodological question of how to collect and use statistics that let us move away from a binary gender division is as yet unsolved. The statistics introduced in this section (real-world data) and the next section (corpus-based data) will therefore be binary gender statistics.

To create our first statistical reference, we retrieved real-world statistics about the distribution of men and women across different professions, from Statistics Sweden (SCB).³ These data were matched against the 43 occupations that occur in our diagnostic sentences. In some cases, we allowed many to one mappings, because the SCB classification was more finegrained than the occupation names in our data. For instance *lärare*

³https://www.statistikdatabasen.scb.se/pxweb/sv/ssd/START__AM__AM0208__AM0208E/YREG50/table/tableViewLayout1/

‘teacher’ in our dataset can be mapped to SCB’s *förskollärare* ‘preschool teacher’, *grundskollärare* ‘primary school teacher’, *gymnasielärare* ‘high school teacher’, *högskolelärare* ‘college teacher’, and *trafiklärare* ‘driving instructor’. In these cases, the SCB statistics were summed together before calculating the female-male ratio. This strategy inevitably influences the results since there is no guarantee that the different SCB occupations have similar female-male ratios.

Looking at our compiled statistics, we see that the occupations in SweWinogender are spread out fairly evenly, covering the whole spectrum from female-dominated (more than two thirds registered female practitioners), through neutral (between one third and two thirds female), to male-dominated professions (less than one third female). Table 1 in Appendix A gives the occupations in SweWinogender sorted after the proportion of registered female practitioners.

4 Gender and occupation as seen from a corpus

Another way to look at occupations as female- or male-dominated is not through work-place statistics, but through the lens of a corpus. We can ask: do people read/write about a certain occupation as associated with men or with women? We could speculate that this correlates much better with preconceptions that people hold than the actual employment statistics. More importantly, however, in the context of evaluating NLP systems: the construction of such systems typically involves corpus data. It therefore makes sense to also investigate the relation between system performance and corpus-based gender and occupation associations.

In Rudinger et al. (2018), the noun gender and number dataset from Bergsma and Lin (2006) is used to this end, cited as a frequently used source of this type of information in actual pronoun resolution systems. This list was created using antecedent-pronoun patterns, defined on an automatically parsed corpus, which were used to extract highly likely cases of co-reference in an unsupervised manner. The proportion of *she* (etc) vs *he* (etc) references to a noun is then used to place it on a scale from feminine to masculine. A counterpart to such a list does not exist for Swedish. Moreover, following the same methodology to create such a list is non-trivial in Swedish: First, it depends on having a parsed corpus of Swedish of

sufficient size and quality. At the time of writing, we have no such corpus readily available. Secondly, several of the patterns used as high-precision coreference patterns by Bergsma and Lin are not useful as sources of information about referential gender in Swedish, because they would involve reflexives or reflexive possessives, which have the same form independent of referential gender.

We therefore follow a less direct approach to extracting occupation-gender associations from corpora, by viewing them as collocative. We assume that a prevalence of definitionally or culturally female-gendered words in the context of mention of a profession, points towards a profession being viewed as female-coded, and correspondingly for male-gendered words. Our approach is reminiscent of the word sense disambiguation method of Yarowsky (1995), and it has been inspired by the application to gendered words in Caren (2013).

As our data source, we use the most recent fifteen years of the Swedish Culturomics Gigaword corpus (Eide et al., 2016), which contains 57M sentences of social media, news text and scientific prose from 2000 to 2015. We use three sets of gendered collocates to classify sentence-level contexts as male- or female-associated: The small set uses only forms of the pronouns *hon/han* ‘he/she’. The medium set also includes a list of definitionally gendered nouns, such as *flicka/pojke* ‘girl/boy’, *mamma/pappa*, *maka/make* ‘wife/husband’, *syster/bror* ‘sister/brother’, etc., in total 31 nouns for the male and 25 nouns for the female set.⁴ In the large set, we include the items from small and medium sets, and in addition a set of culturally gendered items: all female and male proper names with more than 1000 bearers in Sweden.⁵ The large set contains 585 female- and 543 male-gendered words. A sentence is classified according to the majority of collocates it contains – sentences that do not contain any collocates are ignored. For each profession in our dataset, we then calculate the number of sentences classified as female or male that mention this profession. This gives us a way to quantify how strong an occupation is associated with a gender in the corpus.

In many cases, the gender-association assigned

⁴The prototypical pair *man/kvinna* ‘man/woman’ is not included, because *man* ‘man’ is homonymic with the frequent pronoun ‘one’.

⁵This data is also obtained from SCB, at <https://www.statistikdatabasen.scb.se/sq/99310> and <https://www.statistikdatabasen.scb.se/sq/99311>.

to an individual context aligns well with the way an individual referent is presented in the text. This is for instance the case if the decisive collocate in a context happens to be a pronoun that corefers, or is the subject of predication, as in (6) – collocate is in bold, occupation in italics. In these cases, the classification happens to coincide with what Bergsma and Lin’s method would yield.

- (6) Istället blir **han** *börsmäklare* på Wall Street.
‘Instead **he** becomes a *stock broker* on Wall Street.’

But the approach also gives a classification in situations where it makes less sense, for instance in (7), where the context is classified as female because of two collocates from the female set, but where a direct relation to the denotation of the occupation noun is missing.

- (7) **Monica** [...] säger att **hon** hoppas kunna göra en studie för att undersöka hur exempelvis *kassapersonal* påverkas.
‘**Monica** says **she** hopes to be able to study how for example *cashiers* are affected.’

This type of behaviour is to be expected from a collocational approach. As we will see below, comparison of the corpus results to the SCB data suggests that the approach nevertheless yields usable statistics.

In Figure 1 we plot the real-world SCB data against our corpus-derived measure of gender association, for each of the three collocate sets. Irrespective of the collocate set used, our method generally underestimates the female percentages: most points fall below the diagonal in each plot. This effect is clearly stronger in the small collocate set than in the large set. However, this way of looking at the data ignores the fact that the corpora are biased towards classifying sentences as male-associated in general, not just in the context of a profession. The convex curves in the graphs show what a perfect correspondence would look like if we adjust for this corpus-wide bias.⁶ Now we see that in each plot, about half of the points fall above, and half fall below the curve. We conclude that the underestimation of female association of occupations is the result of overall corpus characteristics and not directly related to how people write about

⁶The curves show the line $y = qx / (qx + (1 - q)(1 - x))$, where q is the overall proportion of sentences classified as female in the corpus.

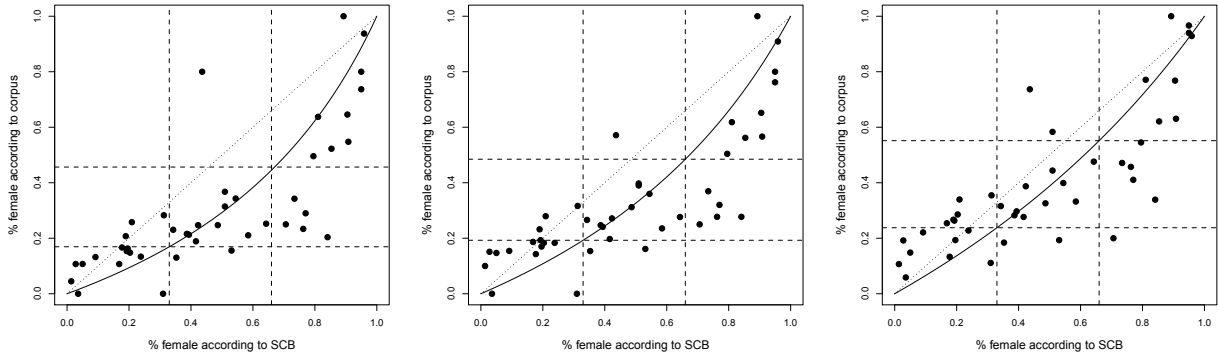


Figure 1: Relation between SCB-based real-world statistics and corpus-based estimations of gender balance in occupations, using small (left), medium (middle), and large collocate sets (right). Curved lines show hypothetical perfect correspondences if we correct for the inherent bias of the method towards male associations. The dashed horizontal lines divide the y-axis in three equal zones male-dominated, neutral, female-dominated, again after correcting for the general bias.

the different professions. However, we can also see a clear pattern in the deviations: points on the left hand-side of the plots generally lie above the curve, whereas those on the right lie below. This means that, compared to the SCB data, the corpus method tends to underestimate male or female domination in the occupations; the estimates shy away from the extremes. This can also be seen by looking at the division of the data into three zones: male-dominated, neutral, and female-dominated. With respect to the SCB data (x-axes) the data points are equally divided between these zones (cf. our remarks in Section 3). However, in the corpus estimates (y-axis), after correction for the overall bias, the neutral professions are over-represented.

On the basis of the overall correlation with the real-world data, we conclude that our method of extracting gender biases for occupations yields meaningful estimates of these biases. We would like to add two further considerations as to why we think our approach makes good sense. Empirically, we note that the pattern that Rudinger et al. (2018) find in the relation between the corpus data and real-world data is (visually) very similar to the patterns discussed above (cf. their figure reproduced here in Appendix B), in spite of what could be expected to be a more precise corpus method. Furthermore, it seems likely that NLP systems that rely on some kind of word embeddings, effectively use collocational information. In those cases, our method may be a much better fit for any biases in such a system than pronoun-resolution-derived estimates.

5 Conclusion

We have presented the freely available SweWinogender test set. It is based on the English Winogender resource and we consider it a starting point which should be expanded upon.

In our data release, the test items themselves will be accompanied by real-world statistics about gender ratios for occupations and by corpus-based gender-occupation associations. These reference data are a core part of making the Winogender idea work as an effective diagnostic.

We have proposed an alternative way of extracting gender-occupation statistics from corpus data, ultimately based on the venerable Distributional Hypothesis. We have argued that the resulting data gives us a perspective on gender and occupation that is relevant to Winogender. Nevertheless, the strengths and weaknesses of this approach need to be further explored. For future work, we will also consider creating further statistical reference sets, for instance in the style of Bergsma and Lin (2006).

We hope that the existence of a SweWinogender will help stimulate the further development, exploration and scrutiny of natural language understanding systems for Swedish.

Acknowledgments

This work has been funded by Nationella Språkbanken – jointly funded by 10 partner institutions and the Swedish Research Council (2018–2024; dnr 2017-00626) – and by the SwedishGlue project (Vinnova, 2020-2021, dnr 2020-02523).

References

- Shane Bergsma and Dekang Lin. 2006. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, page 33–40, USA. Association for Computational Linguistics.
- Yang Trista Cao and Hal Daumé III. 2020. Toward gender-inclusive coreference resolution. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4568–4595, Online. Association for Computational Linguistics.
- Neal Caren. 2013. Using Python to see how the Times writes about men and women. http://nealcaren.github.io/text-as-data/html/times_gender.html. Visited 8 February 2021.
- Stian Rødven Eide, Nina Tahmasebi, and Lars Borin. 2016. The Swedish Culturomics Gigaword corpus: A one billion word Swedish reference dataset for NLP. In *Digital Humanities 2016. From Digitization to Knowledge 2016: Resources and Methods for Semantic Processing of Digital Works/Texts, July 11, 2016, Krakow, Poland*, pages 8–12. Linköping University Electronic Press.
- Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd Schema Challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning, KR’12*, pages 552–561. AAAI Press, Rome, Italy.
- Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2020. Gender bias in neural natural language processing. In *Logic, Language, and Security*, volume 12300 of *Lecture Notes in Computer Science*, pages 189–202. Springer, Cham.
- Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.
- SAOL. 2015. *Svenska akademiens ordlista över svenska språket [The Swedish Academy wordlist of the Swedish language]*, 14th edition. Svenska akademien, Stockholm.
- Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.
- David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, Massachusetts, USA. Association for Computational Linguistics.
- Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Appendix A Occupational Gender Statistics

Occupation	% Female SCB	% Female Corpus	# Corpus Hits
tandhygienist ‘dental hygienist’	95.88	93.75	42
nutritionist ‘nutritionist’	94.97	80.00	90
dietist ‘dietician’	94.97	73.68	250
terapeut ‘therapist’	90.81	54.76	1413
sköterska ‘nurse’	90.51	64.57	3173
juristassistent ‘paralegal’	89.26	100.00	1
frisör ‘hairdresser’	85.36	52.26	868
apotekare ‘pharmacist’	84.10	20.38	404
receptionist ‘receptionist’	81.03	63.74	166
veterinär ‘veterinarian’	79.53	49.57	2139
lärare ‘teacher’	77.02	28.99	16029
bibliotekarie ‘librarian’	76.25	23.37	1061
psykolog ‘psychologist’	73.41	34.25	5078
kassapersonal ‘cashier’	70.62	25.00	5
utredare ‘investigator’	64.27	25.24	2271
revisor ‘accountant’	58.47	21.06	888
läkare ‘physician’	54.43	34.31	16999
kemist ‘chemist’	53.09	15.56	1088
rättsläkare ‘forensic pathologist’	50.95	36.75	205
specialistläkare ‘medical specialist’	50.95	31.43	84
bartender ‘bartender’	48.66	24.71	264
ambulanssjuksköterska ‘paramedic’	43.62	80.00	38
forskare ‘researcher’	42.31	24.71	8070
rådgivare ‘adviser’	41.61	18.90	3253
försäljare ‘sale person’	39.34	21.26	1139
advokat ‘lawyer’	38.67	21.58	10929
arkitekt ‘architect’	35.29	13.02	10744
polis ‘police’	34.26	23.05	47411
bagare ‘baker’	31.25	28.28	361
byggnadsinspektör ‘building inspector’	30.99	0.00	18
ingenjör ‘engineer’	23.82	13.37	4938
operatör ‘operator’	20.92	25.79	851
köksmästare ‘chef’	20.33	14.81	119
programmerare ‘programmer’	19.58	16.30	176
vaktmästare ‘janitor’	19.25	15.38	463
tekniker ‘technician’	18.95	20.74	1080
börsmäklare ‘stockbroker’	17.74	16.67	60
maskinist ‘machine engineer’	16.84	10.71	126
målare ‘painter’	9.16	13.22	3755
mekaniker ‘mechanic’	5.02	10.73	418
vägarbetare ‘road worker’	3.58	0.00	17
elektriker ‘electrician’	2.78	10.71	224
rörmokare ‘plumber’	1.35	4.48	103

Table 1: Occupational Gender Statistics. The smallest set of collocates (only pronouns) was used for the second and third columns

Appendix B Relation between corpus-based noun gender and Bureau of Labor Statistics data

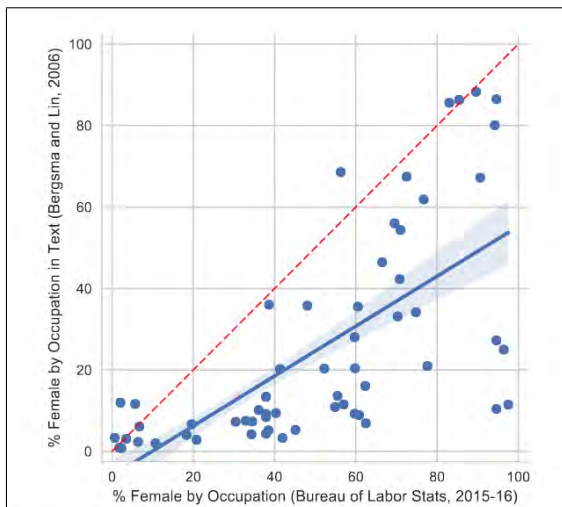


Figure 3: Gender statistics from Bergsma and Lin (2006) correlate with Bureau of Labor Statistics 2015. However, the former has systematically lower female percentages; most points lie well below the 45-degree line (dotted). Regression line and 95% confidence interval in blue. Pearson $r = 0.67$.

Graph and caption reprinted from Rudinger et al. (2018), (c) ACL, CC BY 4.0

Demo Papers

DaNLP: An open-source toolkit for Danish Natural Language Processing

Amalie Brogaard Pauli[♦] Maria Barrett[♦] Ophélie Lacroix[♦] Rasmus Hvingelby^{♦*}

[♦]The Alexandra Institute {amalie.pauli, ophelie.lacroix}@alexandra.dk

[♦]The IT University of Copenhagen mbarrett@itu.dk

[♦]Fraunhofer IIS rasmus.hvingelby@iis.fraunhofer.de

Abstract

We present an open-source toolkit for Danish natural language processing (NLP), enabling easy access to Danish NLP’s latest advancements. The toolkit features wrapper functions for loading models and datasets in a unified way using third-party NLP frameworks. The toolkit is developed to enhance community building, understanding the need from industry and knowledge sharing. As an example of this, we present Angry Tweets: An Annotation Game to increase Danish NLP awareness and create a new sentiment-annotated dataset.

1 Introduction

Danish is the official language in Denmark. It is mainly spoken by the approximately 6M people in Denmark¹. In natural language processing (NLP), Danish is considered a medium-resource language (Joshi et al., 2020). There is, however, limited availability of Danish models and tools (Kirkedal et al., 2019). We believe to increase the availability of NLP resources, we need to engage academia and industry to leverage synergy effects.

In this demonstration paper, we present an open-source, Danish NLP toolkit: *DaNLP*. It contains trained models for named entity recognition (NER), part-of-speech (PoS) tagging, sentiment analysis, parsing, coreference resolution as well as word embeddings and datasets. It is developed in close collaboration with industry and academic partners and aims at strengthening knowledge sharing and community building.

^{*}Rasmus Hvingelby carried out this work while affiliated with the Alexandra Institute.

¹“Danish” refers to standard Danish. The minority languages and dialects of Denmark are not within the scope of this project.

The toolkit makes recent advances in Danish NLP more available and applicable. It is a single entry for accessing Danish NLP resources through a consistent interface. The toolkit consists of resources developed by others and new models and datasets developed within the project guided by what is presently relevant for industry. In the same spirit, we ensure industry-friendly licences, i.e., the resources are licensed for commercial use, ideally without copyleft restrictions. The toolkit employs a unified syntax for loading and applying models inspired by frameworks like scikit-learn (Buitinck et al., 2013) and spaCy (Honnibal et al., 2020).

The overall scope of the DaNLP project is to engage a community of professionals from academia and industry around Danish NLP. As a way of showcasing what is needed concerning annotation and to engage people in the development of Danish NLP tools, a crowdsourcing game is launched as part of the project. This paper’s main contribution is to demonstrate a resource enabling industry’s adoption of NLP for a medium-resource language.

2 Related Work

There are several NLP tools for Danish which we will not review here, but extensive lists exist such as the one by Finn Årup Nielsen.²

We consider an NLP toolkit to be a collection of resources that spans several NLP tasks in one unified framework. This section provides a brief overview of NLP toolkits for Danish and a non-exhaustive selection of comparable languages.

Derczynski et al. (2014) presented an open-source information extraction toolkit for Danish supporting tokenization, named entity recognition (NER) and part-of-speech (PoS) tagging. How-

²<http://www2.imm.dtu.dk/pubdb/edoc/imm6956.pdf>

ever, they are released with a copyleft or non-commercial licence, making them less appealing for industry.

Several multilingual toolkits have some support for Danish, e.g., the Natural Language ToolKit (Loper and Bird, 2002), Polyglot (Al-Rfou et al., 2013), SpaCy (Honnibal et al., 2020), Stanza (Qi et al., 2020), UDPipe (Straka et al., 2016), and Apache OpenNLP (Apache Software Foundation, 2014).

For other medium-resource Germanic languages, language-specific toolkits exist, e.g., Icelandic (Þorsteinsson et al., 2019; Loftsson and Rögnvaldsson, 2007) and Dutch (Bosch et al., 2007; Bouma et al., 2001).

3 The DaNLP toolkit

The DaNLP toolkit contains wrapper functions utilising well-maintained third-party NLP frameworks such as spaCy (Honnibal et al., 2020), Flair (Akbik et al., 2018), Gensim (Řehůřek and Sojka, 2010) and Transformers (Wolf et al., 2020).

The documentation³ for the toolkit provides an overview of the resources with credits to developers, benchmark results, training details, and code snippets for loading and using the models and datasets.

The resources available through the toolkit include both resources developed by others and resources developed specifically as part of the DaNLP project. Therefore, in the following subsections, a † indicates that a resource was created/trained/annotated as part of the DaNLP project. In the opposite case, a reference is supplied.

3.1 Datasets

This subsection provides an overview of available datasets through the DaNLP toolkit.

The Danish Dependency Treebank (DDT) (Buch-Kromann, 2003) consists of texts from the Danish PAROLE corpus (Keson, 1998). The treebank has several layers of annotations but those currently relevant for the models in the toolkit are the *Universal Dependency*(UD) conversion (Johannsen et al., 2015) and the *coreference* annotation. The treebank was additionally annotated with *named entities* and released as the DANE dataset† (Hvingelby et al., 2020).

³<https://danlp-alexandra.readthedocs.io>

NER Besides the DDT annotation (described above), the toolkit also supports the Danish part of WikiANN (Pan et al., 2017) containing Wikipedia articles.

Sentiment Different small sentiment datasets are included: The lcc-sentiment⁴ which contains manual annotated sentences from the Leipzig Corpora Collection (Quasthoff et al., 2006), europarl-da-sentiment⁵, Europarl Sentiment2†, and Twitter Sentiment† described in §4.

Word similarity For evaluating word representations, DaNLP includes two-word similarity datasets: the Danish Similarity Dataset (Schneidermann et al., 2020) and WordSim-353 (Finkelstein et al., 2001).

DanNet (Pedersen et al., 2009), a Danish WordNet (lexical database), is implemented in DaNLP with functions for finding synonyms, hypernyms, etc.

3.2 Models

This section provides an overview of the best performing models⁶ integrated into the toolkit.

NER The best NER model† is fine-tuned on a Danish pre-trained BERT model (Devlin et al., 2019)⁷ and benchmarked on the DaNE annotation† (Hvingelby et al., 2020) using the Transformers architecture from Huggingface (Wolf et al., 2020).

PoS-tagging The best PoS model† implemented in the toolkit is trained using the Flair framework. It is trained and tested on the Danish UD treebank (Johannsen et al., 2015).

Sentiment The toolkit includes sentiment models for three-way polarity†, subjectivity-objectivity detection†, and eight-way emotion detection†. The best polarity and subjectivity-objectivity detection models are trained and benchmarked on Twitter Sentiment† and Europarl Sentiment2† by fine-tuning the Danish BERT model. The polarity models are additionally

⁴<https://github.com/fnielsen/lcc-sentiment>

⁵<https://github.com/fnielsen/europarl-da-sentiment>

⁶based on the most common evaluation metric for the task.

⁷https://github.com/botxo/nordic_bert

benchmarked on lcc-sentiment and europarl-da-sentiment. The Emotion detection model is trained on social media data by fine-tuning the Danish BERT model; however, it was impossible to open-source the data, see §5.

Coreference Resolution The best coreference model[†] is the AllenNLP (Gardner et al., 2018) implementation of Lee et al. (2018) fine-tuned using XLM-Roberta (Conneau et al., 2019) instead of static word embeddings, in line with Joshi et al. (2019). Models are benchmarked on the DDT (Buch-Kromann, 2003).

Dependency Parsing and Chunking We support dependency parsing through the spaCy framework using a model[†] trained on the DDT dataset. We also provide wrapper-code for deducing noun-phrase chunks from predicted dependency trees.

3.3 Text representation

The toolkit contains static word embeddings pre-trained by third-parties⁸ with word2vec (Bojanowski et al., 2017) and fastText (Mikolov et al., 2013). Dynamic word embeddings[†], trained using the Flair architecture (Akbik et al., 2018) are also available in the toolkit, as well as embeddings derived from the Danish BERT language model.⁹

3.4 DaNLP: Selected examples of usage

The goal of the DaNLP project is to make datasets and models easily accessible through a unified syntax. Therefore, the package provides consistent functions for loading datasets through prominent frameworks such as spaCy or Flair – e.g., for training purposes – or in standard datatypes or formats such as DataFrames¹⁰ or CoNLL-U¹¹. Below is an example of several possibilities for loading the DDT:

```
#Danish Dependency Treebank
from danlp.datasets import DDT
ddt = DDT()
spacy_corpus= ddt.load_with_spacy()
flair_corpus = ddt.load_with_flair()
conllu_format = ddt.load_as_conllu()
```

⁸<https://loar.kb.dk/handle/1902/329>,
<https://fasttext.cc/docs/en/crawl-vectors.html>, <https://github.com/danish-stance-detectors/RumourResolution>

⁹https://github.com/botxo/nordic_bert

¹⁰<https://pandas.pydata.org/>

¹¹<https://universaldependencies.org/format.html>

Models can also be loaded with a unified syntax. However, there are differences in applying them based on the framework they are trained with, though most of them are provided with simple prediction functions that take a sentence as input. Below is an example of how to load and use the Emotion detection model:

```
# Emotion Detection
from danlp.models import (
    load_bert_emotion_model )
clf = load_bert_emotion_model()
clf.predict("Jeg ser frem til det")
```

4 Angry Tweets: An Annotation Game

To advance the field of Danish NLP, there is a need for task-specific annotated corpora for training and benchmarking models. (Kirkedal et al., 2019; Sprognævn, 2019). The DaNLP project has previously annotated a corpus using a traditional approach, i.e., with a few trained annotators. However, such annotations are expensive and time-consuming. Therefore, we propose collaborative crowdsourcing, designed as a game. A similar approach is previously seen in Öhman et al. (2018). Like their gamified emotion annotation setup, we also asked participants to annotate a few gold-annotated sentences as well as sentences previously annotated by other crowd annotators in order to assess the annotation. The main motivation, besides creating a new Twitter sentiment corpus for the toolkit, was engaging professionals and other people interested in the development of Danish NLP and communicating what is needed in terms of annotation work. The game was, therefore, announced through social media, a blog post, and the Danish medium Data Tech.¹² The hope was to motivate volunteer participants to contribute to the development of Danish NLP. The gamification element (failing is an option, and there was also a possibility to win a symbolic prize) is meant to peak people’s interest and motivate them to supply high-quality annotations. We made an effort to keep a light and fun tone with a storyline including a swan, the project logo.

The game interface The game consists of eight rounds with four tweets per round. Figure 1 shows

¹²<https://pro.ing.dk/datatech/article/angry-tweets-vaer-med-til-bygge-dataset-over-foelelsesladede-tweets-9496>

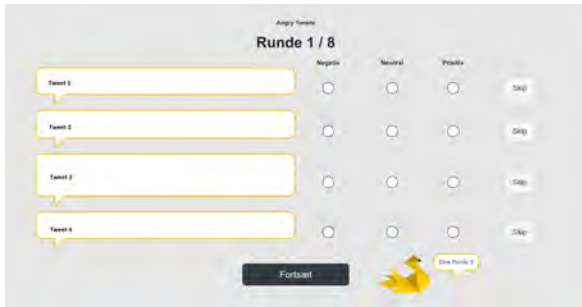


Figure 1: One annotation page in Angry Tweets.

one round. The tweets are annotated with three-class sentiment (positive, neutral, negative). As a part of a defensive task design (Sabou et al., 2014), participants were on every second page asked to annotate one gold-annotated tweet, and on each round, the completion time was measured. Not passing any of these checks triggered game over. In each round, participants annotated one sentence already annotated by another annotator and was rewarded with a point if their annotation matched the previous annotation.

Statistics The game was completed 114 times. 82% of players completing the game submitted a contact email to participate in the competition for a prize, indicating that some participants were not motivated by the prize. The tweets are collected through a list of commonly used Danish hashtags and posted between January and May 2019. The corpus consists of 4921 annotated tweets, where 1266 is double annotated with an inter-annotator agreement of 65%. The majority is annotated through the game, but 1727 was annotated by one trained annotator.

5 Knowledge In Knowledge Out

The development of DaNLP is industry-focused. Therefore, the DaNLP team is in dialogue with Danish companies and government agencies to understand their needs. The project also shares knowledge and disseminates.

Throughout the project, there have been dialogues with around 35 companies consisting of both start-ups and larger tech companies, as well as eight different government agencies. There is a large spread in the maturity of using and understanding NLP across organisations. Some companies are pushing the field of Danish NLP forward, and their requests are generally more data; large, raw text corpora and annotated corpora. Other

companies are new to the field and mostly driven by curiosity, and a third category consists of companies with a more task-oriented desire. Here, we especially noted a need for better performing models for NER and sentiment analysis. Therefore, these tasks were the initial focus of the toolkit. To stay in close contact with industry, two collaborations with companies were constituted: One with a media monitoring company, Infomedia A/S, to improve their existing NER system for news articles. The other collaboration was with the Danish Broadcasting Corporation to monitor the mood on their social media platform.

The knowledge-sharing part is aimed at making more people and companies aware of the possibilities of NLP. Therefore the project includes a blog on Danish NLP¹³, NLP introduction talks, and a demonstration page to show some of the models in action.¹⁴

6 A community for Danish NLP

The toolkit has so far benefited from bug reports, bug fixes, and suggestions for improvements from contributors through our GitHub repository.¹⁵ The ambition is to have an even stronger community contributing to the toolkit with new models and datasets. The ambition is that the toolkit in time becomes more community-driven.

It is also within the project’s scope to contribute to NLP frameworks to enable Danish’s direct support. Before DaNLP, spaCy did not support Danish since an open-source NE dataset was lacking. However, with DaNE, (Hvingelby et al., 2020) this was fulfilled and is now part of spaCy.¹⁶ The Flair tagging models for PoS and NER trained as part of DaNLP are now also available directly through Flair.¹⁷

Nevertheless, the need for improving Danish NLP goes beyond a toolkit. It seems like the timing is opportune; currently, parties from academia and industry in Denmark are starting collaboration. Examples are recent, open-source models released by companies¹⁸ and a large cross-

¹³<https://medium.com/danlp>

¹⁴<https://danlp-demo.alexandra.dk>

¹⁵<https://github.com/alexandrainst/danlp>

¹⁶<https://spacy.io/models/da>

¹⁷https://github.com/flairNLP/flair/blob/master/resources/docs/TUTORIAL_2_TAGGING.md

¹⁸<https://github.com/sarnikowski/>

collaboration on a large Danish text corpus named Gigaword by Strømberg-Derczynski et al. (2020). To strengthen the community around Danish NLP, the DaNLP project have gathered both companies in front of the field and researchers from Danish Universities (the Danish Technical University, the University of Copenhagen, and the IT University of Copenhagen) for network meetings to discuss and collaborate on Danish NLP¹⁹. One of the major identified challenges is how to gather and share data safely concerning privacy and GDPR.²⁰

7 Concluding remarks

DaNLP is a new toolkit to make Danish NLP more applicable to industry. With this aim, the DaNLP project has been engaged in dialogues with industry, knowledge sharing and community building with academia and professionals. The hope is to continue working with a stronger community and inspire similar projects in other low to medium resource languages.

Acknowledgments

We want to thank the DaNLP team and Leon Derczynski, and our collaborators: the Danish Broadcasting Corporation and Infomedia. This work is supported by two performance contracts funded by the Danish Ministry of Higher Education and Science: “Dansk for Alle” and “Digital sikkerhed, etik og dataetik”. Maria Barrett is supported by a research grant (34437) from VILLUM FONDEN.

References

- Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.
- Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.
- danish_transformers/tree/main/electra
https://github.com/botxo/nordic_bert
¹⁹To join: <https://danlp.alexandra.dk/>
²⁰<https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN>
- Apache Software Foundation. 2014. OpenNLP Natural Language Processing Library. <Http://opennlp.apache.org/>.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Antal van den Bosch, Bertjan Busser, Sander Canisius, and Walter Daelemans. 2007. An efficient memory-based morphosyntactic tagger and parser for Dutch. *LOT Occasional Series*, 7:191–206.
- Gosse Bouma, Gertjan van Noord, and Robert Malouf. 2001. Alpino: Wide-coverage computational analysis of Dutch. In *Computational Linguistics in the Netherlands 2000*, pages 45–59. Brill Rodopi.
- Matthias Buch-Kromann. 2003. The Danish Dependency Treebank and the DTAG treebank tool. In *2nd Workshop on Treebanks and Linguistic Theories (TLT), Sweden*, pages 217–220.
- Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.
- Leon Derczynski, Camilla Vilhelmsen Field, and Kenneth S Bøgh. 2014. DKIE: Open source information extraction for Danish. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 61–64.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppín. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414.

- Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python.
- Rasmus Hvingelby, Amalie Brogaard Pauli, Maria Barrett, Christina Rosted, Lasse Malm Lidegaard, and Anders Søgaard. 2020. DaNE: A named entity resource for Danish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4597–4604. European Language Resources Association.
- Anders Johannsen, Héctor Martínez Alonso, and Barbara Plank. 2015. Universal dependencies for Danish. In *International Workshop on Treebanks and Linguistic Theories (TLT14)*, page 157.
- Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel S Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5807–5812.
- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293.
- Britt Keson. 1998. Vejledning til det danske morfosyntaktisk taggedede parole-korpus. *Parole report, Det Danske Sprog- og Litteraturselskab (DSL)*.
- Andreas Kirkedal, Barbara Plank, Leon Derczynski, and Natalie Schluter. 2019. The lacunae of Danish natural language processing. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 356–362.
- Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692.
- Hrafn Loftsson and Eiríkur Rögnvaldsson. 2007. IceNLP: A natural language processing toolkit for Icelandic. In *Eighth Annual Conference of the International Speech Communication Association*.
- Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26:3111–3119.
- Emily Öhman, Kaisla Kajava, Jörg Tiedemann, and Timo Honkela. 2018. Creating a dataset for multilingual fine-grained emotion-detection using gamification-based annotation. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 24–30.
- Vilhjálmur Þorsteinsson, Hulda Óladóttir, and Hrafn Loftsson. 2019. A wide-coverage context-free grammar for Icelandic and an accompanying parsing system. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1397–1404.
- Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958. Association for Computational Linguistics.
- Bolette Sandford Pedersen, Sanni Nimb, Jörg Asmussen, Nicolai Hartvig Sørensen, Lars Trap-Jensen, and Henrik Lorentzen. 2009. DanNet: the challenge of compiling a wordnet for Danish by reusing a monolingual dictionary. *Language resources and evaluation*, 43(3):269–299.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.
- Uwe Quasthoff, Matthias Richter, and Christian Bie-mann. 2006. Corpus portal for search in monolingual corpora. In *LREC*, pages 1799–1802.
- Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.
- Marta Sabou, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. 2014. Corpus annotation through crowdsourcing: Towards best practice guidelines. In *LREC 2014*, pages 859–866.
- Nina Schneidermann, Rasmus Hvingelby, and Bolette Sandford Pedersen. 2020. Towards a gold standard for evaluating Danish word embeddings. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4754–4763.

Dansk Sprognævn. 2019. Dansk sprogteknologi i verdensklasse.

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipes: trainable pipeline for processing conllu files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297.

Leon Strømberg-Derczynski, Rebekah Baglini, Morten H Christiansen, Manuel R Ciosici, Jacob Aarup Dalsgaard, Riccardo Fusaroli, Peter Juel Henriksen, Rasmus Hvingelby, Andreas Kirkedal, Alex Speed Kjeldsen, et al. 2020. The Danish gigaword project. *arXiv preprint arXiv:2005.03521*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

HB Deid - HB De-identification tool demonstrator

Hercules Dalianis

Department of Computer
and Systems Sciences
Stockholm University
Kista, Sweden
hercules@dsv.su.se

Hanna Berg

Department of Computer
and Systems Sciences
Stockholm University
Kista, Sweden
hanna@hidetext.se

Abstract

This paper describes a freely available web-based demonstrator called HB Deid. HB Deid identifies so-called protected health information, PHI, in a text written in Swedish and removes, masks, or replaces them with surrogates or pseudonyms. PHIs are named entities such as personal names, locations, ages, phone numbers, dates. HB Deid uses a CRF model trained on non-sensitive annotated text in Swedish, as well as a rule-based post-processing step for finding PHI. The final step in obscuring the PHI is then to either mask it, show only the class name or use a rule-based pseudonymisation system to replace it.

1 Introduction

Electronic patient records and other health data contain information that can identify a patient. These data need to be *washed*, for both legal and ethical reasons before they can be re-used for various purposes as research or machine learning.

This paper presents a machine learning-based demonstrator called HB Deid, Health Bank De-Identification tool. It is a tool for automatic de-identification and pseudonymisation of protected health information, PHI. PHIs are information that can identify a patient. PHIs are similar to named entities and encompass for example *personal names, locations, phone numbers, dates, ages*, etc. PHIs can be present both in structured and unstructured clinical data such as free text. In structured data PHIs are easily identifiable, the table information informs of the category of the data the whole table can be removed or obscured. In the free clinical text there is a greater effort to identify

a PHI since, for example, the PHI entity *Parkinson* is difficult to classify whether it is a disease name or a personal name. Similarly, *Sjögren's* in *Sjögren's syndrome* may be mistaken for a patient's name. In this paper, we focus on PHIs¹ in free text in electronic patient records.

A demonstrator is a pedagogical instrument to show a system or an idea and let a broader audience explore - in our case - a research system or pilot system.

The process of de-identifying text typically uses two steps. Firstly, a PHI is identified through named entity recognition. Secondly, the PHI is obscured. Strategies for hiding information may be masking the information or replacing it with a surrogate through a process called pseudonymisation. Pseudonymisation makes the text more fluent to read and when also removing the annotation tags it inconceives the identification of potentially remaining sensitive data in plain sight and protects the identification of PHIs. This method is called Hiding in Plain Sight (HIPS), (Carrell et al., 2012).

A substantial amount of studies have been published on the de-identification of text (Meystre et al., 2010; Stubbs et al., 2015). While most of these studies have focused on English, research have been carried out for French (Grouin and Névéol, 2014), Spanish (Marimon et al., 2019), Danish (Pantazos et al., 2016) and Swedish (Berg and Dalianis, 2019) as well as Japanese (Kajiyama et al., 2020).

Generally, high recall is preferred over high precision in de-identification research as the privacy of the individuals describe is of paramount importance. It is therefore important to not miss any sensitive information.

With regards to pseudonymisation, there are fewer studies. One of the first was a study by Sweeney (1996), which described a system for

¹HideText, <http://www.hidetext.se> is the platform where HB Deid is commercialised.

¹PHI, Personally identifying information, is a more general term which includes other domains.

identifying PHI and then replacing them with surrogates, but not how this process was carried out. In another study by Douglass et al. (2004) this pseudonymisation process is described elaborated such as that dates were shifted, personal names were shifted to other personal names in the Boston area. Locations were shifted randomly, and hospitals were given fictitious names.

For de-identification and pseudonymisation for English there is yet another study (Deleger et al., 2014). For pseudonymisation for Swedish, there is a system described in (Dalianis, 2019).

While there are plenty of demonstrators of Named Entity Recognition systems the only available one for de-identification, to our knowledge, is the HitzalMed demonstrator (Lopez et al., 2020). The HitzalMed demonstrator is constructed for de-identification and pseudonymisation of Spanish electronic patient records. To try it out a registration must be carried out here². The system identifies and categorizes entities into different categories, as: *first name, last name, location, phone number, age, date and health care unit*. While the system is intended for Spanish, this part works relatively well for English and Swedish. The system then either masks or replaces the sensitive information with surrogates - which are in Spanish.

2 The HB Deid demonstrator

The HB Deid demonstrator³, see Figure 1, is an attempt to show the possibilities of de-identification and pseudonymisation techniques for electronic patient records written in Swedish. The system is based on the work carried out by Berg and Dalianis (2019) for de-identification and by Dalianis (2019) for the pseudonymisation.

The data used to train HB Deid is not the original set of annotated sensitive electronic patient records in Swedish - the Stockholm EPR PHI Corpus, but the corpus is indirectly used since its trained model is used to improve the public available Swedish news corpora called *Webbnyheter 2012* that are semi-manually annotated for the NER classes PER and LOC and ORG and MISC⁴. *Webbnyheter 2012* was machine annotated using the original sensitive trained model from Stockholm EPR PHI Corpus and then it was corrected

²HitzalMed registration, <https://snlt.vicomtech.org/hitzalmed/demo/help>

³HB Deid, <http://hbdeid.dsv.su.se>

⁴Swedish NER Corpus, <https://github.com/klintan/swedish-ner-corpus>

manually, using both the manual annotations and the machine annotations to decide on the correct annotation. This effort was carried out to avoid using the sensitive Stockholm EPR PHI Corpus directly.

The bootstrapped model in the HB Deid demonstrator has not yet been evaluated, but the original sensitive model has been evaluated and the results reported in (Berg and Dalianis, 2019).

2.1 The HB Deid classes

The HB Deid demonstrator identifies the following PHI classes: *First name, last name, location, phone number, age, date, health care unit, organization and personal number (social security number)*.

2.2 Programming languages and machine learning environment

HB Deid is developed in the programming language Python and is machine learning-based with a rule-based post-processing step, (Berg and Dalianis, 2019). It uses the CRF Conditional Random Fields algorithm (Lafferty et al., 2001) as implemented in CRFSuite (Okazaki, 2007) with a `sklearn-crfsuite` wrapper⁵.

The pseudonymiser is completely rule-based and uses dictionaries to generate surrogates in a fashion similar to (Dalianis, 2019). Compared to the pseudonymiser in (Dalianis, 2019) personal names are replaced with greater variation. While common names are replaced with other common names, uncommon names are replaced with uncommon names. The uncommon names in the dictionaries are 123,000 female first names, 121,000 male first names and 35,000 last names.

The web interface for the HB-demo is written in Flask⁶ that in turn is coded in Python.

2.3 Interface

The Flask web interface for HB Deid uses encryption according to the HTTPS protocol. Nothing processed is saved on the web-server.

See Figure 1 for all the possible menu choices and below for a detailed description:

- *Ersättare - Replacer*. Decides the shapes of the processed text, see the following choices.

⁵`sklearn-crfsuite`, <https://github.com/TeamHG-Memex/sklearn-crfsuite>

⁶Flask, <https://flask.palletsprojects.com/en/1.1.x/>

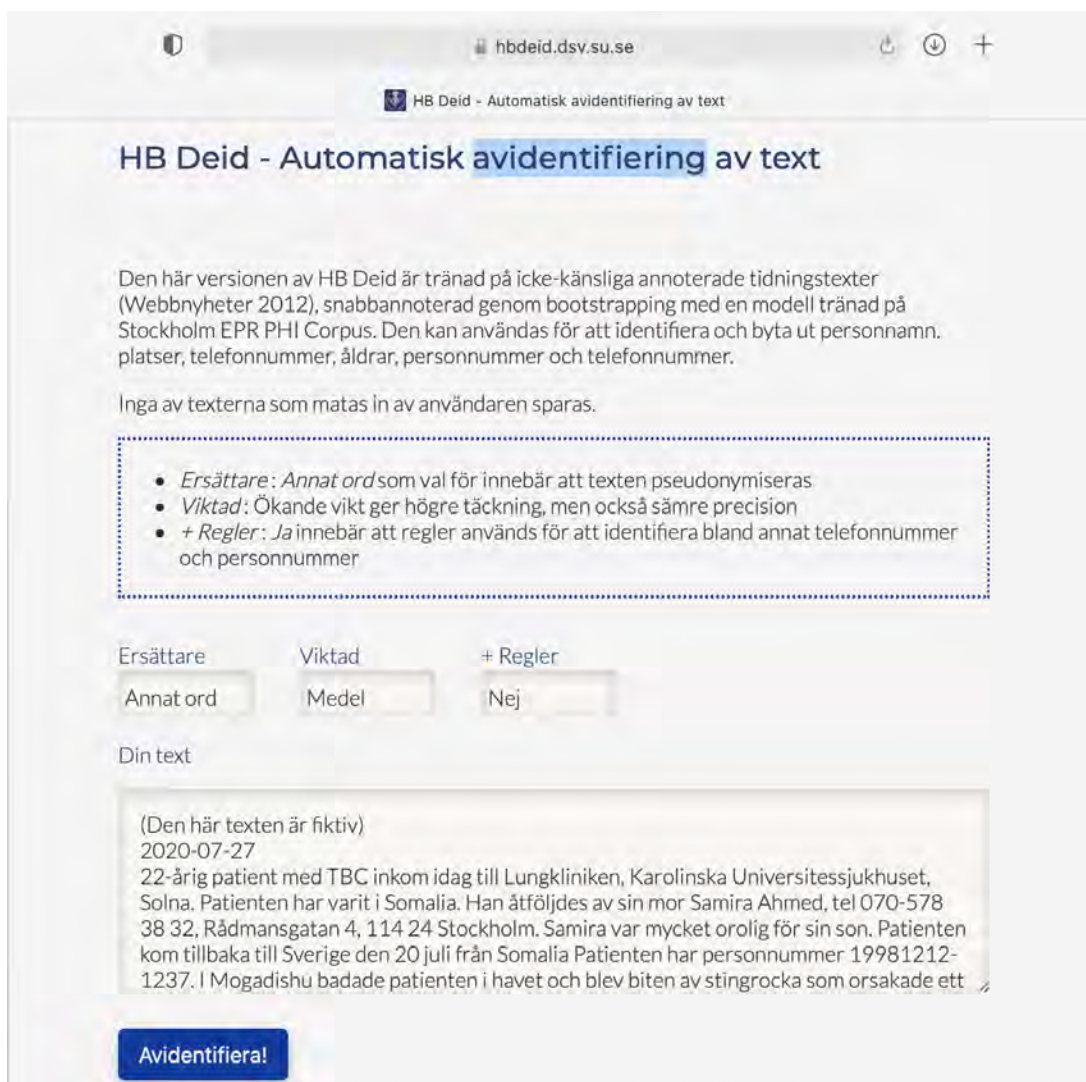


Figure 1: The interface of HB Deid (In Swedish) with the various choices. The text shown is fictitious.

- *Ersätt inte* - Do not replace. Tag the PHI with the class name.
- *Annat ord* - Other word. Replace the PHI with a surrogate or pseudonym.
- *Klass* - Class. Replace the PHI with the class name.
- *Mask*. Mask the PHI with XXX.
- *Vikter* - Weights. Increases the recall in three steps.
- *+ Regler* - Rules uses a post-processing step utilising rules, mainly controlling the output from the machine learning tool but also uses regular expressions to find personal numbers and phone numbers.

The output from HB Deid can be seen in Figure 2.

The interface and the functionality of the HB Deid demonstrator have not been evaluated yet, since one bottleneck is that the demonstrator must either be installed at the hospital or be set up inside the Health Bank⁷ infrastructure laboratory environment at the university to be evaluated by clinicians.

3 Conclusion

We have shown how to train and construct an automatic de-identification and pseudonymisation tool for clinical text in Swedish. We have also used a bootstrapped language model that is privacy protected. Finally, we have made a user friendly and freely available web interface to the demonstrator called HB Deid.

The demonstrator has been presented in the

⁷Health Bank, <https://dsv.su.se/healthbank>

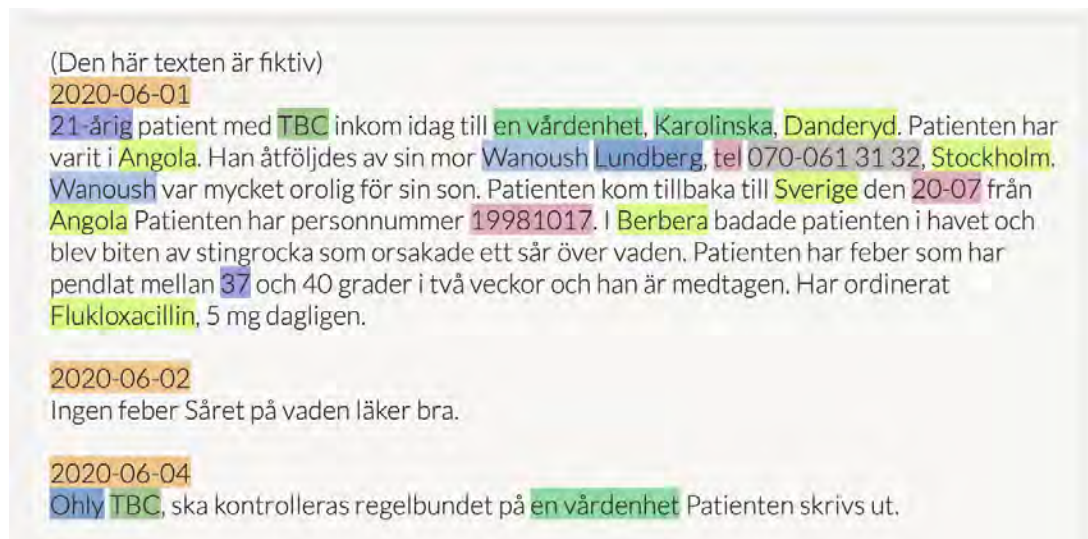


Figure 2: The output from HB Deid in form of a de-identified and pseudonymised text. The various colours represent the different classes. Moving the mouse pointer over an identified coloured entity will display the corresponding class name.

Swedish trade magazine Computer Sweden, (Lindström, 2020).

We have also been contacted by many users that had asked us about HB Deid and given us feedback on how to improve the system. They have also asked us if they can use HB Deid for other purposes as de-identification of transcribed interviews.

One more proposal was to have the possibility to correct wrong predictions by re-annotate them directly in the HB Deid interface to re-learn HB Deid.

We plan to let Stockholm Regional Council and Karolinska University Hospital test HB Deid on their electronic patient record texts with the future aim to de-identify them before they are handed out to researchers or for machine learning purposes.

Acknowledgements

We are grateful to the DataLEASH project for funding this research work.

References

Hanna Berg and Hercules Dalianis. 2019. Augmenting a de-identification system for Swedish clinical text using open resources and deep learning. In *Proceedings of the Workshop on NLP and Pseudonymisation*, pages 8–15, Turku, Finland. Linköping Electronic Press.

David Carrell, Bradley Malin, John Aberdeen, Samuel Bayer, Cheryl Clark, Ben Wellner, and Lynette

Hirschman. 2012. Hiding in plain sight: use of realistic surrogates to reduce exposure of protected health information in clinical text. *Journal of the American Medical Informatics Association*, 20(2):342–348.

Hercules Dalianis. 2019. Pseudonymisation of Swedish electronic patient records using a rule-based approach. In *Proceedings of the Workshop on NLP and Pseudonymisation*, pages 16–23, Turku, Finland. Linköping Electronic Press.

Louise Deleger, Todd Lingren, Yizhao Ni, Megan Kaiser, Laura Stoutenborough, Keith Marsolo, Michal Kouril, Katalin Molnar, and Imre Solti. 2014. Preparing an annotated gold standard corpus to share with extramural investigators for de-identification research. *Journal of Biomedical Informatics*, 50:173–183.

Margaret Douglass, Gari D. Clifford, Andrew Reisner, George B. Moody, and Roger G. Mark. 2004. Computer-assisted de-identification of free text in the MIMIC II database. In *Computers in Cardiology, 2004*, pages 341–344. IEEE.

Cyril Grouin and Aurélie Névéol. 2014. De-identification of clinical notes in french: towards a protocol for reference corpus development. *Journal of biomedical informatics*, 50:151–161.

Kohei Kajiyama, Hiromasa Horiguchi, Takashi Okumura, Mizuki Morita, and Yoshinobu Kano. 2020. De-identifying free text of japanese electronic health records. *Journal of Biomedical Semantics*, 11(1):1–12.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling se-

- quence data. In *Proceedings 18th International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann.
- Karin Lindström. 2020. AI förbereder för AI – tvättar bort känsliga uppgifter ur texter (In Swedish), <https://computersweden.idg.se/2.2683/1.744319/ai-tvattar-kansliga-uppgifter>. *Computer Sweden*.
- Salvador Lima Lopez, Naiara Perez, Laura García-Sardiña, and Montse Cuadros. 2020. HitzalMed: Anonymisation of Clinical Text in Spanish. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, May 13-15, Marseille*, pages 7038–7043.
- Montserrat Marimon, Aitor Gonzalez-Agirre, Ander Intxaurre, Heidi Rodriguez, Jose Lopez Martin, Marta Villegas, and Martin Krallinger. 2019. Automatic De-identification of Medical Texts in Spanish: the MEDDOCAN Track, Corpus, Guidelines, Methods and Evaluation of Results. In *IberLEF@SEPLN, Sociedad Española para el Procesamiento del Lenguaje Natural*, pages 618–638.
- Stephane M Meystre, F Jeffrey Friedlin, Brett R South, Shuying Shen, and Matthew H Samore. 2010. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC medical research methodology*, 10(1):70.
- Naoaki Okazaki. 2007. CRFsuite: a fast implementation of Conditional Random Fields, <http://www.chokkan.org/software/crfsuite>. Accessed 2021-02-02.
- Kostas Pantazos, Søren Lauesen, and Søren Lippert. 2016. Preserving medical correctness, readability and consistency in de-identified health records. *Health Informatics Journal*, page 1460458216647760.
- Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. 2015. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/UTHealth shared task Track 1. *Journal of Biomedical Informatics*, 58:S11–S19.
- Latanya Sweeney. 1996. Replacing personally-identifying information in medical records, the scrub system. In *Proceedings of the AMIA annual fall symposium*, page 333. American Medical Informatics Association.

