Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018)

Dag Haug, Stephan Oepen, Lilja Øvrelid,
Marie Candito, and Jan Hajič (Editors)

December 13–14, 2018

Oslo University (Norway)

Published by

Linköping University Electronic Press, Sweden

Linköping Electronic Conference Proceedings #155

(ISSN 1650-3740; eISSN 1650-3740; ISBN 978-91-7685-137-1)

Preface

The tradition of annual international workshops on Treebanks and Linguistic Theories (TLT) dates back to 2002, when the series was first launched in Sozopol, Bulgaria. Part of this proud tradition is to host TLT in Norway in years after the workshop was held in Prague, Czech Republic. With a sense of tradition as well as pride, this volume comprises the proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018), held on the campus of the University of Oslo, Norway, on December 13 and 14, 2018.

TLT addresses all aspects of treebank design, development, and use. As 'treebanks' we consider any pairing of natural language data (spoken or written) with annotations of linguistic structure at various levels of analysis, ranging from e.g. morpho-phonology to discourse. Annotations can take any form (including trees or general graphs), but they should be encoded in a way that enables computational processing. Reflecting growing community interest in natural language understanding, TLT 2018 includes a thematic session on the interface between syntax and semantics and on 'meaning banking' and its applications.

The worshop received 21 submissions from all over Europe and the US (and one each from Brazil and Japan), of which 15 are collected in this volume and will be presented at the conference. All submissions were reviewed by at least three experts in the field, and the final selection was made by the Programme Committee. We are indebted to everyone who contributed to the reviewing and selection process. The conference programme is complemented by two invited keynotes by distinguished researchers from Poland and The Netherlands, as well as by a soul-searching panel discussion on the topic

Syntactico-Semantic Representations in Natural Language Processing:
Vital, Venerable, or Vacuous?

TLT 2018 is made possible by the joint work of many dedicated individuals, in particular the Programme Committee and the TLT Steering Committee (Jan Hajič, Erhard Hinrichs, Sandra Kübler, Joakim Nivre, and Petya Osenova); we warmly acknowledge their enthusiasm and community spirit. We are grateful to the Department of Informatics and to the Department of Linguistics and Scandinavian Studies at the University of Oslo for generously making available financial support, infrastructure, and staff time. The workshop is financially supported by the Center for Advanced Studies (CAS) at the Norwegian Academy of Science and Letters, who have co-located a meeting of their international research group *SynSem: From Form to Meaning* and, thus, make an important contribution to keeping participation fees at quite reasonable levels (by Norwegian standards).

With not quite one week to go, we expect some 40–45 participants at the workshop and much look forward to welcoming our colleagues and peers to Oslo.

Dan Haug, Stephan Oepen, Lilja Øvrelid (Local Organizers)

Programme Chairs

- Marie Candito, Université Paris Diderot, France
- Jan Hajič, Charles University in Prague, Czech Republic
- Dag Haug, University of Oslo, Norway
- Stephan Oepen, University of Oslo, Norway
- Lilja Øvrelid, University of Oslo, Norway

Organizing Committee

- Dag Haug
- Stephan Oepen
- Lilja Øvrelid

Programme Committee

- Sandra Kübler, Indiana University Bloomington, USA
- Jan Hajič, Charles University in Prague, Czech Republic
- Zdenka Uresova, Charles University in Prague, Czech Republic
- Marie Candito, Universtité Paris 7, France
- Teresa Lynn, Dublin City University, Ireland
- Agnieszka Patejuk, Polish Academy of Sciences, Poland
- Lori Levin, Carnegie Mellon University, USA
- Joakim Nivre, Uppsala University, Sweden
- Koenraad De Smedt, University of Bergen, Norway
- Olga Scrivner, Indiana University Bloomington, USA
- Djamé Seddah, Université Paris la Sorbonne, France
- Patricia Amaral, Indiana University Bloomington, USA
- Daniel Zeman, Charles University in Prague, Czech Republic
- Heike Zinsmeister, University of Hamburg, Germany
- Ann Bies, Linguistic Data Consortium, USA
- Petya Osenova, Sofia University, Bulgaria
- Memduh Gökırmak, Istanbul Technical University, Turkey

- Kaili Müürisep, University of Tartu, Estonia
- Emily M. Bender, University of Washington, USA
- Silvie Cinková, Charles University in Prague, Czech Republic
- Jiří Mírovský, Charles University in Prague, Czech Republic
- Tatjana Scheffler, Universität Potsdam, Germany
- Barbora Hladka, Charles University in Prague, Czech Republic
- Eva Hajičova, Charles University in Prague, Czech Republic
- Marie-Catherine de Marneffe, The Ohio State University, USA
- Eckhard Bick, University of Southern Denmark, Denmark
- Gosse Bouma, Rijksuniversiteit Groningen, The Netherlands

Table of Contents

Invited Keynotes

Malvina Nissim
Using Weak Signal in NLP **1**

Adam Przepiórkowski
Coordination in Universal Dependencies **3**

Regular Papers

Tatiana Bladier, Andreas van Cranenburgh, Kilian Evang, Laura Kallmeyer, Robin Möllemann, Rainer Osswald
RRGbank: a Role and Reference Grammar Corpus of Syntactic Structures Extracted from the Penn Treebank **5**

Gosse Bouma
Comparing Two Methods for Adding Enhanced Dependencies to UD Treebanks **17**

Peter Bourgonje, Manfred Stede
The Potsdam Commentary Corpus 2.1 in ANNIS3 **31**

Alastair Butler Stephen Wright Horn
Parsed Annotation with Semantic Calculation **39**

Kira Droganova, Olga Lyashevskaya, Daniel Zeman
Data Conversion and Consistency of Monolingual Corpora: Russian UD Treebanks **53**

Dan Flickinger
Measuring Evolution of Implemented Grammars **67**

Eva Fučíková, Eva Hajičová, Jan Hajič, Zdeňka Urešová
Defining Verbal Synonyms: Between Syntax and Semantics **75**

Matías Guzmán Naranjo, Laura Becker
Quantitative Word Order Typology with UD **91**

Jirka Hana, Barbora Hladka
Universal Dependencies and a Non-Native Czech **105**

Sardar Jaf, Thomas G. Hudson
On the Development of a Large Scale Corpus for Native Language Identification **115**

Sonja Marković, Daniel Zeman
Reflexives in Universal Dependencies **131**

Eleni Metheniti, Günter Neumann
Wikinflection: Massive Semi-Supervised Generation of Multilingual Inflectional Corpus from Wiktionary **147**

Noor Abo Mokh, Sandra Kübler
Preprocessing Does Matter: Parsing Non-Segmented Arabic **163**

Atreyee Mukherjee, Sandra Kübler
Domain Adaptation in Dependency Parsing via Transformation Based Error Driven Learning **179**

Tobias Pütz, Daniël de Kok, Sebastian Pütz, Erhard Hinrichs
Seq2Seq or Perceptrons for Robust Lemmatization. An Empirical Examination **193**

Using Weak Signal in NLP

Malvina Nissim
University of Groningen
m.nissim@rug.nl

ABSTRACT
Treebanking requires substantial expert labour in the annotation of a variety of language phenomena. Possibly to a lesser extent for phenomena where laypeople can also contribute, the need for assigning manual labels nevertheless characterises almost all language processing tasks, since they are usually best solved by supervised models. Such models are indeed accurate, but we also know that they lack portability, as they are bound to languages, genres, and even specific datasets. Having spent years dealing with annotation issues and label acquisition for various semantic and pragmatic tasks, in this talk I take a radically different perspective, which hopefully can yield interesting reflections over treebanking, too. I will show various ways to cheaply obtain and exploit weaker signal in supervised learning, even venturing on the suggestion to reduce existing strong, accurate signal in order to enhance portability. I will do so via discussing three case studies in three different classification tasks, all focused on social media.

KEYWORDS: supervised learning, weak signal, social media.

Coordination in Universal Dependencies

Adam Przepiórkowski

Polish Academy of Sciences and University of Warsaw

adamp@ipipan.waw.pl

ABSTRACT

The aim of this talk is to review theoretical dependency approaches to coordination and to propose an extension to the Universal Dependencies (UD) treatment of this phenomenon. Some emphasis will be put on aspects of coordination which are currently problematic for UD, namely, nested coordination and coordination of unlike grammatical functions.

KEYWORDS: coordination, dependency syntax, Universal Dependencies.

RRGbank: a Role and Reference Grammar Corpus of Syntactic Structures Extracted from the Penn Treebank

Tatiana Bladier[1], *Andreas van Cranenburgh*[2], *Kilian Evang*[1], *Laura Kallmeyer*[1], *Robin Möllemann*[1], *Rainer Osswald*[1]

(1) University of Düsseldorf, Germany
(2) University of Groningen, the Netherlands

{bladier,evang,kallmeyer,moellemann,osswald}@phil.hhu.de,
a.w.van.cranenburgh@rug.nl

ABSTRACT

This paper presents RRGbank, a corpus of syntactic trees from the Penn Treebank automatically converted to syntactic structures following Role and Reference Grammar (RRG). RRGbank is the first large linguistic resource in the RRG community and can be used in data-driven and data-oriented downstream linguistic applications. We show challenges encountered while converting PTB trees to RRG structures, introduce our annotation tool, and evaluate the automatic conversion process.

KEYWORDS: Role and Reference Grammar, RRG, treebank conversion, Penn Treebank.

1 Introduction

Wide empirical coverage is a touchstone for every grammatical theory. Treebanks have been widely used as training material for data-driven parsing approaches, data-oriented language processing, statistical linguistic studies, or machine learning throughout the last decades. However, no large linguistic resource exists for the framework of Role and Reference Grammar (RRG; Van Valin and LaPolla, 1997; Van Valin, 2005) so far. In this paper we describe the development of the first annotated corpus of RRG structures[1] created through (semi-)automatic conversion of the Penn Treebank.

Providing a treebank resource to the RRG community will be useful for several reasons: (i) it will be a valuable resource for corpus-based investigations in the context of linguistic modeling using RRG and in the context of formalizing RRG, which is needed for a precise understanding of the theory and for using it in NLP contexts. Efforts towards a formalization of RRG as a tree-rewriting grammar have already been made recently (Kallmeyer et al., 2013; Kallmeyer, 2016; Kallmeyer and Osswald, 2017). (ii) In the context of implementing precision grammars, at least for English, an RRG treebank is useful for testing the grammar and evaluating its coverage. (iii) It will enable supervised data-driven approaches to RRG parsing (grammar induction and probabilistic parsing). (iv) Finally, and more immediately, the specification of the treebank transformation yields valuable new insights into RRG analyses of English syntax — since, even though RRG has covered a large range of typologically different languages, compared to other theories, English has not been considered much.

Since manual annotation is very time-consuming, we decided to (semi-)automatically derive RRGbank from an existing treebank. For this, we chose the Penn Treebank (PTB; Marcus et al., 1993) because of its large size and and availability of additional layers such as OntoNotes (Hovy et al., 2006) which may be used to enrich RRGbank in the future. The PTB has been used in the past, among others, for deriving CCGbank, a corpus of Combinatory Categorial Grammar derivations (Hockenmaier and Steedman, 2007). We decided to start from the original PTB rather than CCGbank because its phrase structure trees are more similar to RRG than CCG derivations, and to avoid possible compounding of errors in automatic conversion. A different route to creating treebanks is taken by the LinGO Redwoods and ParGram approaches to dynamic treebanking for HPSG and LFG, respectively (Oepen et al., 2004; Flickinger et al., 2012; Sulger et al., 2013). These projects made use of manually developed grammars and parsers for the grammar formalisms in question, and then manually checked and selected the best output among all possible outputs. This is not an option for RRGbank at the moment because no wide-coverage computational grammar for RRG is available yet, but it may be a possible avenue in the future, after such a grammar has been extracted from RRGbank.

2 Syntactic Structures in Role and Reference Grammar

2.1 Brief Overview of RRG

RRG is intended to serve as an explanatory theory of grammar as well as a descriptive framework for field researchers. It is a functional theory of grammar which is strongly inspired by typological concerns and which aims at integrating syntactic, semantic and pragmatic levels of description (Van Valin, 2005, 2010). In RRG, there is a direct mapping between the semantic and syntactic representations of a sentence, unmediated by any kind of abstract syntactic representations. In particular, RRG is a strictly non-transformational theory and therefore does not make use of

[1] A demo version of the treebank is available at rrgbank.phil.hhu.de.

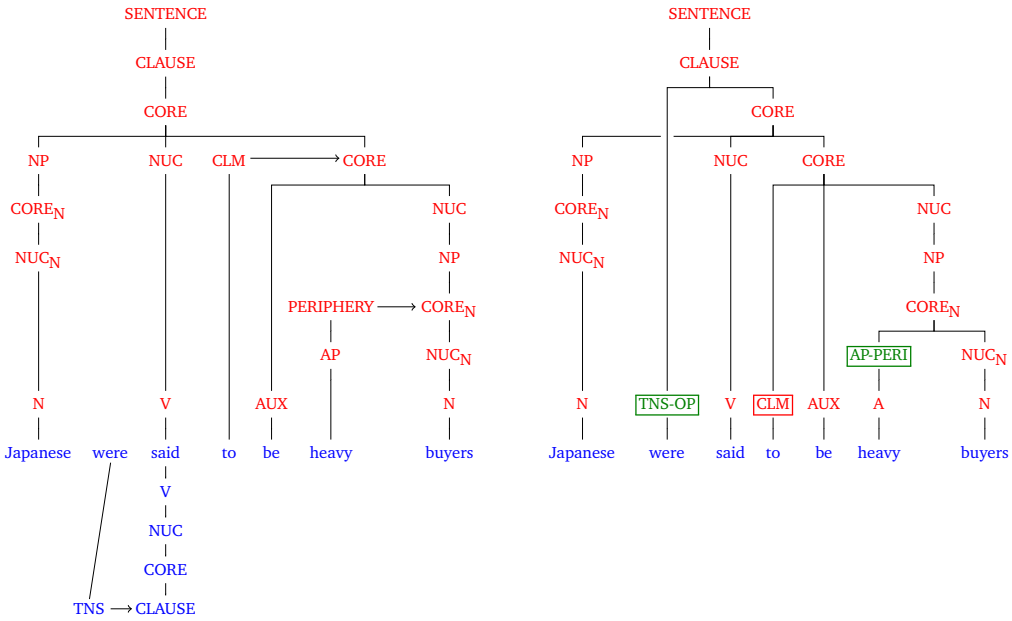


Figure 1: Representation of periphery, operator projection and clause-linkage-markers (CLMs) in standard RRG structures (left-hand side) and our notational variant (right-hand side).

traces and the like; there is only a single syntactic representation for a sentence that corresponds to its actual form. The mapping between the syntactic and semantic representations is subject to an elaborate system of linking constraints. For the purposes of the present paper, only the syntactic side of the representations is taken into account.

A key assumption of the RRG approach to syntactic analysis is a *layered structure* of the clause: The *core* layer consists of the *nucleus*, which specifies the (verbal) predicate, and its arguments. The *clause* layer contains the *core* plus extracted arguments, and each of the layers can have a *periphery* for attaching adjuncts (as shown for example in Figure 1). Another important feature of RRG is the separate representation of *operators*, which are closed-class morphosyntactic elements for encoding tense, modality, aspect, etc. Operators attach to those layers over which they take semantic scope. Since the surface order of the operators relative to arguments and adjuncts is much less transparent and often requires crossing branches, RRG represents the constituent structure and the operator structure as different *projections* of the clause (usually drawn above and below the sentence, respectively).

2.2 Tree Annotation Format for RRG Syntactic Structures

The standard data structure for constituent treebank annotations is trees, specifically, a single tree per sentence whose leaves are the tokens and whose structure and constituent and edge labels depend on the concrete annotation scheme. Many computational tools that process and use treebanks, such as query engines and parsers, rely on this format. By contrast, the usual notation for RRG syntactic structures departs from it in two ways (cf. Van Valin, 2005, 2010). Firstly, there are *two* trees per sentence, the constituent projection and the operator projection. A second idiosyncratic element is the use of arrows (instead of edges) for attaching peripheral

constituents (adjuncts) and clause linkage markers (CLMs), as well as the operators in the operator projection.

To resolve this discrepancy, we adopt a notational variant in which each RRG structure is represented as a single tree, exemplified in the right half of Figure 1. Firstly, note that the spine of the operator projection always mirrors that of the constituent projection. We thus simply identify the corresponding nodes (such as the CLAUSE, CORE, NUC and V nodes in the example) and attach operators in the same tree as other constituents. Secondly, we represent arrows as ordinary edges (and eliminate PERIPHERY nodes), whereby the roots of operators, peripheries and clause linkage markers become daughters of the nodes they attach to (see the TNS, CLM and AP nodes in the example). In order to still distinguish operators and peripheries, we decorate the labels of their roots with -OP and -PERI, respectively. Clause linkage markers are already distinguished by the root label CLM. As a result, we obtain trees that sometimes have crossing branches, resulting from operator scope (see Figure 1 on the right) or from adjunct scope (see Figure 2).

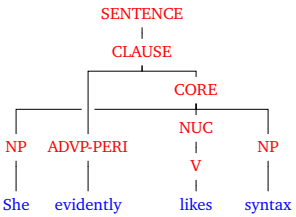


Figure 2: Periphery with crossing branches in RRG.

3 From Penn Treebank to RRGbank

We transform PTB annotations into RRG annotations by iteratively combining automatic conversion with manual correction. The process is sketched in Figure 3. We started with a small sample of sentences from the PTB ($n = 16$). Annotators with RRG expertise annotated these sentences from scratch with RRG trees, without looking at the PTB annotation, resulting in a small validation treebank. We then developed a conversion algorithm which transforms PTB trees into RRG trees. This development was *error-driven*, that is, the algorithm was improved step by step until its output was identical to the gold standard annotation.

We then used the developed algorithm to convert a larger sample ($n = 100$) of PTB trees to RRG.[2] The resulting "silver-standard" annotation was checked and corrected by annotators, using a click/drag/drop-based interface we developed, shown in Figure 7.[3] Correcting silver-standard data is less time-consuming than annotating from scratch; thus in this way we were able to increase the size of our validation treebank iteratively. After this step the set of conversion rules was updated again in order to correctly convert the entire new set of sentences. We plan to repeat the process of manual tree correction and updating the set of conversion rules to increase it further.

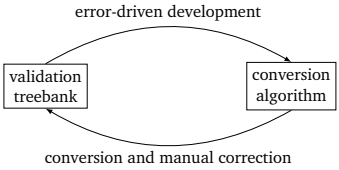


Figure 3: Annotation through iterative conversion and correction.

In the following subsections, we motivate and describe the conversion algorithm in more detail.

[2]The sentences were selected randomly from Sections 02–21 of the PTB, but we excluded sentences that contained fragmentary constituents (marked FRAG) or were longer than 25 tokens.

[3]See rrgbank.phil.hhu.de for a set of demo sentences.

3.1 Differences between PTB Trees and RRG Structures

We illustrate some important differences between PTB and RRG syntactic structures in Figure 4: First, the PTB assumes a separate VP projection inside clauses which does not include the subject, whereas RRG groups the subject together with other arguments in the *core*. This is due to RRG's semantic approach to argument realization. Second, while the PTB treats auxiliaries similarly to other verbs, RRG treats them as operators and attaches them according to their semantic scope. Copulas are the exception to this, as RRG attaches them within the *core*, signalling the following element to be the *nucleus*.

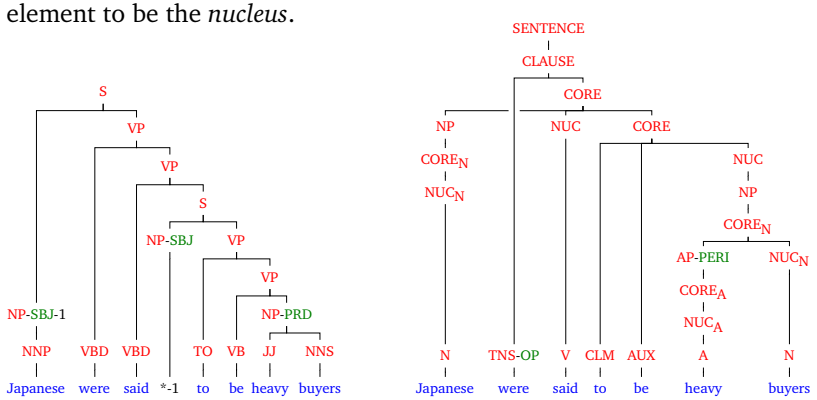


Figure 4: An example of a sentence from PTB (left tree) converted to RRG (right tree).

Third, the PTB uses *traces* to mark non-local dependencies whereas RRG has no such notion (the trace and the corresponding constituent in the PTB are marked with numbers, as shown in Figure 4 on the left-hand side). Fourth, adjuncts and other non-arguments like the adjective *heavy* in the example are analyzed as peripheries in RRG. Note that attachment of operators (as in Figure 4) and peripheries (as in Figure 2) according to their semantic scope can lead to crossing branches in RRG structures, which never occur in the PTB. Figure 5 shows the rules which were used for the conversion.

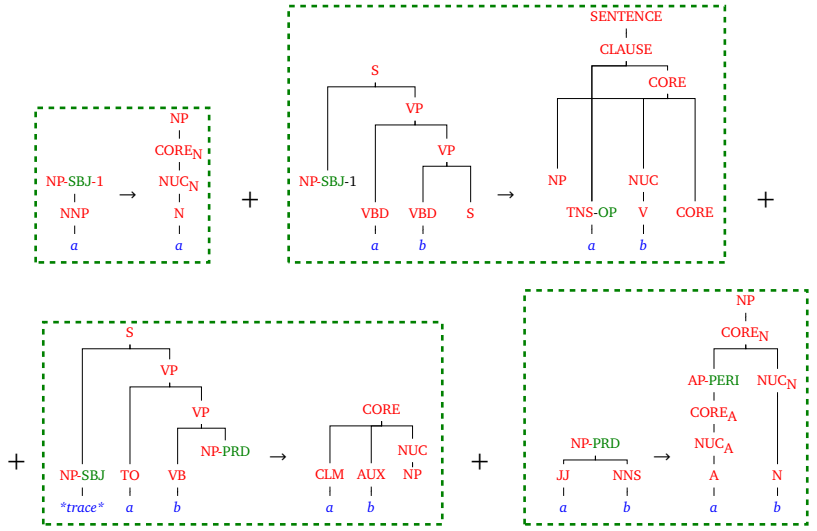


Figure 5: Conversion rules used for the sentence from Figure 4.

3.2 Outline of the Conversion Algorithm

The conversion algorithm was developed in an error-driven way, as outlined above. To each tree, the algorithm applies a series of rules. Each rule applies to specific constituents and may introduce, remove and relabel nodes. We started this conversion process by defining rules for the most frequent constituent types, with the aim of covering the whole treebank.

3.2.1 Conversion Algorithm: Regular Transformation Rules

In order to convert the PTB trees to RRG structures we created a relatively small set of general transformation rules applicable to all constituents of the same type throughout the PTB corpus. Some of these rules convert constituents with exactly one child node (Figure 6a). Other rules are used to convert larger constituents. For example, the rule in Figure 6b rewrites a basic sentence with a transitive verb to an RRG structure. Figure 6c shows one of the rules for transforming topicalized constituents to a left-detached position (LDP) in RRG.

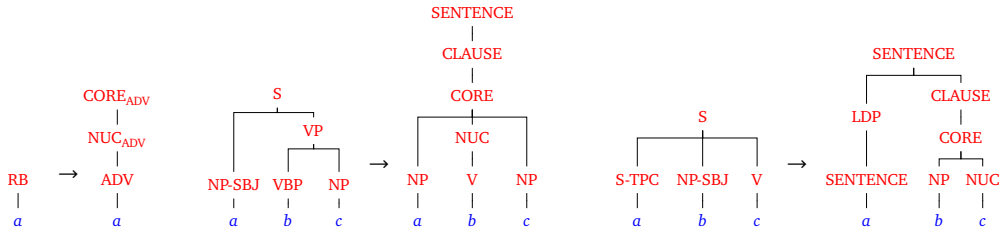


Figure 6: Three examples of conversion rules for PTB trees.

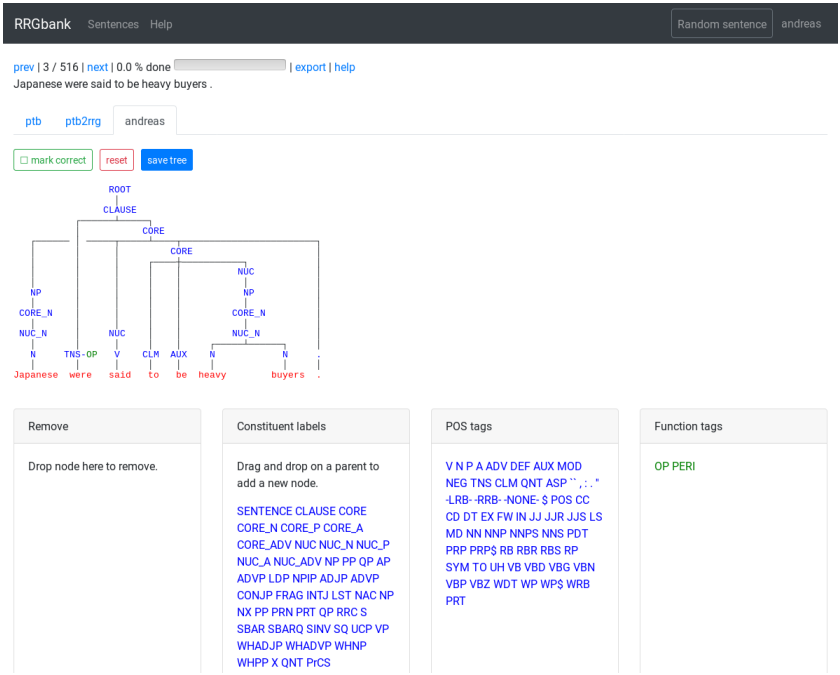


Figure 7: The annotation interface.

3.2.2 Problematic Cases for Conversion

The majority of the constituents in the PTB can be transformed with a small set of transformation rules, described in the previous section. However, the conversion process also revealed some systematic sources of conversion mistakes, among which are the following.

Annotation inconsistencies or errors in the PTB. In the example in Figure 8, a noun *network* is erroneously annotated as a verb. In such cases of annotation inconsistencies in the PTB, we do not introduce special conversions rules, since they would become too specific and only applicable for this particular sentence.

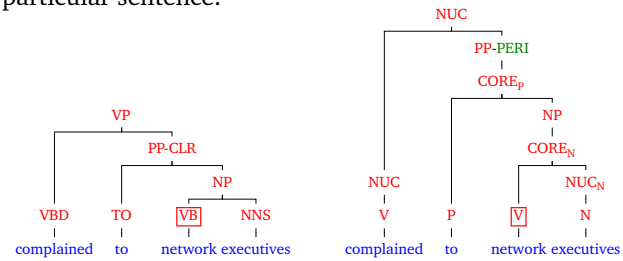


Figure 8: Errors in the PTB annotation.

Underspecific annotation in the PTB. In some cases, a deterministic conversion from PTB to RRG annotations is not possible because RRG makes distinctions that the PTB does not (always) make. One case in point is the negation operator *not*, which is always attached as an adverb inside a VP in the PTB, but can be attached to different layers in RRG depending on its semantic scope (see Figures 9). The RRG analysis provided in the middle tree on Figure 9 displays the case of internal negation with the possible readings "Japan is not a political country (but Belgium is)" or also "Japan is not a political country (it is a cultural one)". External negation however, negates the proposition as a whole, so the sentence displayed in the right tree in Figure 9 can be read as "It is not the case, that Japan is a political country".

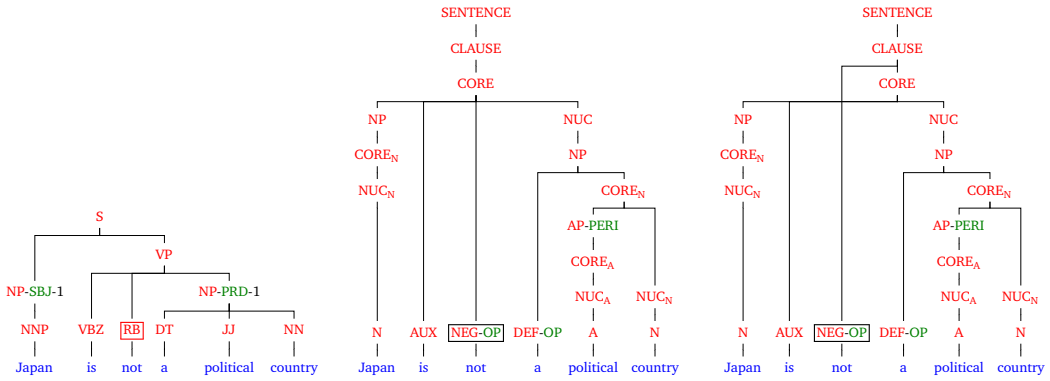


Figure 9: Difficult constructions in RRG: scope of negation in the PTB and in RRG.

Moreover, the trees in Penn Treebank and RRG structures are not deterministically related. That is, similar tree structures in the PTB might require different analyses in RRG. Figures 10 and 11 display the difference between two juncture types in RRG. Figure 10 shows the case of *core cosubordination*, in which the cores share their operators, while operator sharing is not required for *coordinated cores* (Figure 11).

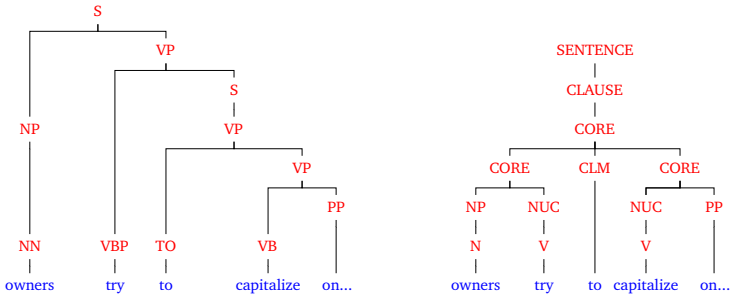


Figure 10: Core cosubordination.

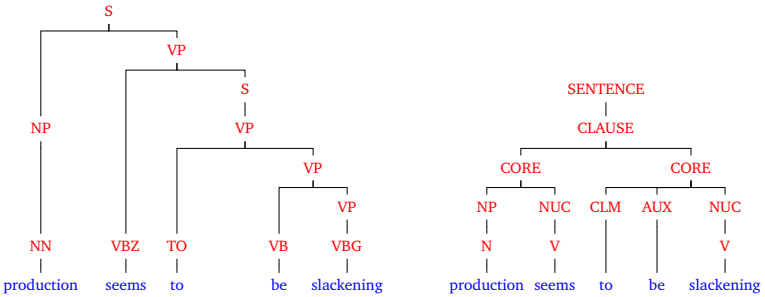


Figure 11: Core coordination.

RRG also differentiates between restrictive and non-restrictive relative clauses (see Figures 12 and 13). Restrictive relative clauses restrict the possible referents of the modified nominal expression by specifying information about them.

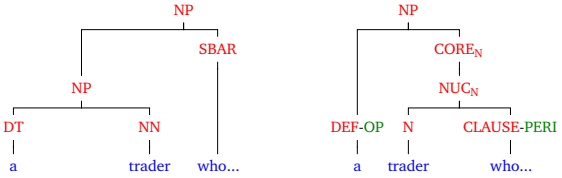


Figure 12: Restrictive relative clause.

Non-restrictive relative clauses, usually separated by a comma, encode additional information about a referent which is already unambiguously identifiable.

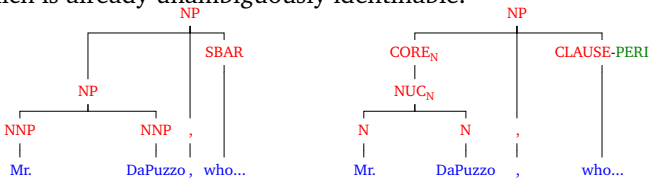


Figure 13: Non-restrictive relative clause.

Another example of underspecification in the Penn Treebank is the distinction between argument (non-pheripheral) PPs, which are to be labeled PP, and adjunct (peripheral) PPs, which are to be labeled PP-PERI. In some cases, functional labels in the PTB (for example, PP-TMP for temporal PPs or PP-DIR for directional PPs) indicate adjuncthood, while in other cases, the PTB provides

no such marking (compare, for example, the PP attachments in Figures 8 and 14).

Open questions in the theory of RRG. The process of converting PTB trees to RRG structures also reveals a number of under-investigated issues within RRG. An example is treatment of quantifier phrases (QPs). In particular, the PTB treats various kinds of constituents as QPs which can be headed by different lexical categories. The analysis of quantifiers differs in RRG, where some elements are analyzed as operators and others as peripheries. In such cases, we decided to leave problematic constituents unchanged until sufficient linguistic analysis is provided (see Figure 14).

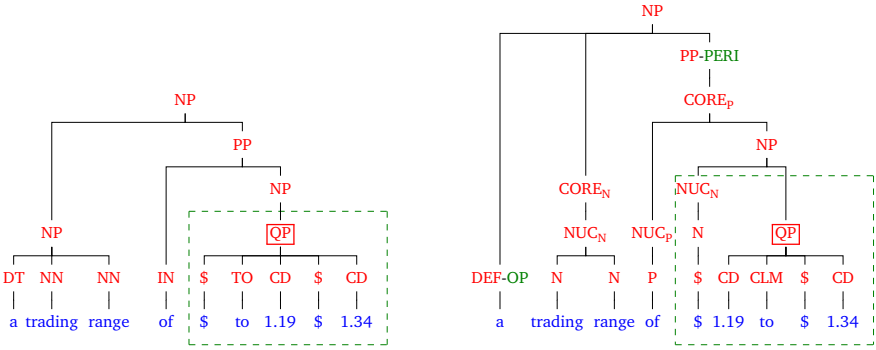


Figure 14: An open question in RRG: Quantifier phrases (marked with dashed lines).

4 Evaluation

We evaluate our conversion algorithm in terms of *completeness* and *correctness*.

Our algorithm finds an output tree for every input tree from the Penn Treebank. We measure the *completeness* of conversion as the ratio of nodes in a tree that have a label in the RRG label set. Because the PTB and RRG share some labels (e.g., NP, PP), this measure is nonzero even before conversion. Applied to WSJ Sections 02–21 of the Penn Treebank, completeness is currently 25.0% before conversion and 97.1% after conversion.

To measure *correctness*, we apply the algorithm to our validation treebank. This currently contains 100 RRG structures that have been manually corrected by one annotator. We are in the process of increasing this number to at least 500 and repeating the correction process with a second annotator to compute inter-annotator agreement and perform arbitration. In Table 1, we provide a preliminary evaluation of our conversion algorithm by comparing its output to the 100 corrected structures. We measure correctness in terms of shared labeled bracketings (the EVALB measure) of the automatic output and the annotated test set.

We also evaluated our conversion algorithm on different constituents since some of them turned out to be more problematic for the automatic conversion than the others. Table 1 provides an overview of the conversion scores for different constituents. Among the most problematic rewriting rules are those which are used to convert the constituents to highly complex structures in the framework of RRG (for example, CORE, NUC or CORE_N). These structures can include different elements and exhibit different arrangements of these elements (compare, for example, the RRG structures in Figures 1, 2 and 8). By contrast, constituents such as CORE_A or NUC_ADV tend to be non-problematic for the conversion since their structure is either highly predictable (CORE_A (A)) or is clearly indicated by the corresponding labels in the PTB (for example, ADVP

label	frequency	recall	precision	F1
(any)	100.00	91.18	90.21	90.69
NP	14.74	96.04	95.40	95.72
CORE_N	14.48	90.36	89.16	89.76
NUC_N	13.89	91.36	86.31	88.76
CORE	6.49	75.00	77.32	76.14
NUC	6.49	87.50	87.06	87.28
CLAUSE	5.19	78.75	86.90	82.62
NUC_P	5.16	100.00	98.15	99.07
PP	5.13	97.47	96.86	97.16
CORE_P	5.13	97.47	96.86	97.16
AP	3.80	90.60	92.17	91.38
CORE_A	3.73	93.91	93.10	93.51
NUC_A	3.73	97.39	96.55	96.97
ROOT	3.25	100.00	100.00	100.00
ADVP	2.30	81.69	96.67	88.55
NUC_ADV	2.21	100.00	95.77	97.84
CORE_ADV	2.21	92.65	88.73	90.60

Table 1: Preliminary results of evaluating the conversion algorithm on our 100-sentence validation corpus, overall and for the 15 most frequent constituent labels. The scores are labeled EVALB scores.

for adverbial phrases).

5 Conclusion

This paper reports on ongoing efforts towards creating a treebank for Role and Reference Grammar, a grammar theory that is widely used in typological research and that adopts a view on grammar as a complex system of syntax, semantics, morhpology, and information structure. We concentrate on the syntactic analyses assumed in RRG, and we first proposed a tree-based representation structure for them. We then started an iterative process of annotating PTB sentences with RRG structures, developing rules for an automatic transformation of PTB trees into RRG trees, and then feeding back information about errors on the gold data into the development of transformation rules. We plan to continue this cycle of annotation, rule development and testing for some time.

The work presented here will lead to RRGbank, an RRG annotation of the PTB. RRGbank will be the first large linguistic resource in the RRG community. It opens up new possibilities for using RRG in natural language processing (grammar implementation, grammar induction, data-driven parsing, semantic parsing when adding for instance the semantic information from PropBank etc.). Furthermore, the development of RRGbank will also lead to new insights about how to analyze certain constructions in English within RRG, and the treebank will be a valuable resource for empirical, corpus-based investigations of RRG structures.

We also plan to explore treebanks available in the framework of the Universal Dependencies project (Nivre et al., 2016) for conversion to RRG structures. An advantage of using Universal Dependencies is the coverage of many languages along with a uniform labeling while taking into consideration linguistic peculiarities of each language.

The transformation tool will be made available and, in addition, we plan to provide RRGbank via the Linguistic Data Consortium (LDC) as an alternative annotation layer to the PTB.

Acknowledgments

The work presented in this paper was partly funded by the European Research Council (ERC grant TreeGraSP) and partly by the German Science Foundation (CRC 991). We would also like to thank Robert D. Van Valin, Jr. for giving us valuable advice for our project. Furthermore, we are grateful to three anonymous reviewers whose comments helped to improve the paper.

References

Flickinger, D., Kordoni, V., and Zhang, Y. (2012). DeepBank: A Dynamically Annotated Treebank of the Wall Street Journal. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories*, Lisbon, Portugal.

Hockenmaier, J. and Steedman, M. (2007). CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3).

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60. Association for Computational Linguistics.

Kallmeyer, L. (2016). On the mild context-sensitivity of *k*-Tree Wrapping Grammar. In Foret, A., Morrill, G., Muskens, R., Osswald, R., and Pogodalla, S., editors, *Formal Grammar: 20th and 21st International Conferences, FG 2015, Barcelona, Spain, August 2015, Revised Selected Papers. FG 2016, Bozen, Italy, August 2016, Proceedings*, number 9804 in Lecture Notes in Computer Science, pages 77–93, Berlin. Springer.

Kallmeyer, L. and Osswald, R. (2017). Combining Predicate-Argument Structure and Operator Projection: Clause Structure in Role and Reference Grammar. In *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 61–70, Umeå, Sweden. Association for Computational Linguistics.

Kallmeyer, L., Osswald, R., and Van Valin, Jr., R. D. (2013). Tree Wrapping for Role and Reference Grammar. In Morrill, G. and Nederhof, M.-J., editors, *Formal Grammar 2012/2013*, volume 8036 of *LNCS*, pages 175–190. Springer.

Marcus, M., Santorini, B., and Marcinkiewicz, M. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Nivre, J., De Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R. T., Petrov, S., Pyysalo, S., Silveira, N., et al. (2016). Universal dependencies v1: A multilingual treebank collection. In *LREC*.

Oepen, S., Flickinger, D., Toutanova, K., and Manning, C. D. (2004). Lingo redwoods. *Research on Language and Computation*, 2(4):575–596.

Sulger, S., Butt, M., King, T. H., Meurer, P., Laczkó, T., Rákosi, G., Dione, C. B., Dyvik, H., Rosén, V., De Smedt, K., et al. (2013). Pargrambank: The pargram parallel treebank. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 550–560.

Van Valin, Jr., R. D. (2005). *Exploring the Syntax-Semantics Interface*. Cambridge University Press.

Van Valin, Jr., R. D. (2010). Role and Reference Grammar as a framework for linguistic analysis. In Heine, B. and Narrog, H., editors, *The Oxford Handbook of Linguistic Analysis*, pages 703–738. Oxford University Press, Oxford.

Van Valin, Jr., R. D. and LaPolla, R. (1997). *Syntax: Structure, meaning and function*. Cambridge University Press.

Comparing two methods for adding Enhanced Dependencies to UD treebanks

Gosse Bouma[1,2]
(1) Center for Language and Cognition
University of Groningen
(2) Center for Advanced Study
Norwegian Academy for Science and Letters
g.bouma@rug.nl

ABSTRACT
When adding enhanced dependencies to an existing UD treebank, one can opt for heuristics that predict the enhanced dependencies on the basis of the UD annotation only. If the treebank is the result of conversion from an underlying treebank, an alternative is to produce the enhanced dependencies directly on the basis of this underlying annotation. Here we present a method for doing the latter for the Dutch UD treebanks. We compare our method with the UD-based approach of Schuster et al. (2018). While there are a number of systematic differences in the output of both methods, it appears these are the result of insufficient detail in the annotation guidelines and it is not the case that one approach is superior over the other in principle.

KEYWORDS: Universal Dependencies, Enhanced Universal Dependencies, Dutch, Ellipsis, Coordination, Control.

1 Introduction

While universal dependencies (UD) have proven to be a convenient level of syntactic annotation for many applications, such as language typology (Futrell et al., 2015), stylistics (Wang and Liu, 2017), (cross-lingual) parser comparison (Zeman et al., 2017), relation extraction (Shwartz et al., 2016), and construction of word embeddings (Vulić, 2017), there is also the concern that it is not capturing all relevant syntactic distinctions. The majority of the current treebanks in the UD repository[1] are converted from underlying treebanks that were annotated with a language specific annotation scheme.[2] A frequent observation is that some of the distinctions in the underlying treebank cannot be preserved in the conversion to UD (Lipenkova and Souček, 2014; Pyysalo et al., 2015; Przepiórkowski and Patejuk, 2018). Enhanced universal dependencies (EUD) (see below) can help to diminish the loss of information, especially as it offers the means to annotate syntactic control relations and a more principled solution to annotating elliptical constructions.

EUD should also make it easier to obtain fully specified semantic interpretations from UD annotated text, as explored in Reddy et al. (2017) and Gotham and Haug (2018). Raising and control are syntactic phenomena that influence the semantic interpretation of a sentence. Computing the correct control relations on the basis of UD alone is cumbersome, and may require access to additional lexical resources. Similarly, the semantic interpretation of ellipsis requires identification of the elided material. The UD annotation of ellipsis avoids reference to empty elements, and thus cannot provide all information required for interpreting elliptical constituents. EUD does allow empty nodes, and thus provides a better basis for semantic interpretation.

1.1 Enhanced Universal Dependencies

Enhanced universal dependencies[3] are motivated by the need to obtain a syntactic annotation layer that is more suitable for semantic tasks, such as textual entailment, information extraction, or deriving a full semantic representation of an utterance. In particular, it proposes the following enhancements to basic UD:

- null nodes for elided predicates,
- propagation of conjuncts,
- additional subject relations for control and raising constructions,
- coreference in relative clause constructions.
- modifier labels that contain the preposition or other case-marking information
- conjunction labels that contain the coordinating word.

The EUD guidelines follow the proposals in Schuster and Manning (2016) and the proposed analysis of ellipsis in Schuster et al. (2017).

[1] `http://universaldependencies.org/`

[2] The documentation for 17 of the 102 treebanks in release v2.1 states that dependency relations were manually added, while 59 treebanks contain dependency relations that were created by converting an underlying (manually annotated) treebank.

[3] `http://universaldependencies.org/u/overview/enhanced-syntax.html`

It is clear that the EUD annotation of ellipsis, as well as the addition of control relations, cannot be done on the basis of information contained in the UD annotation only, and thus we present our method for doing this on the basis of the underlying annotation below. Propagation of conjuncts as defined in the guidelines appears to be possible on the basis of the basic UD annotation itself. Nevertheless, we include it in the discussion below, as our treatment is more comprehensive than what is suggested by the guidelines, and also because it interacts with the reconstruction of elided predicates and the addition of control relations. The analysis of relative clauses in EUD adds a dependency from the head of the relative clause to the antecedent noun. The label of this newly introduced dependency is identical to the label of the dependency from the head of the relative clause to the relative pronoun in UD. Thus, this extension can be computed on the basis of UD alone, as long as a relative pronoun is present. However, we do note some challenging cases in the next section. The last two modifications (fine-grained labels for *nmod*, *obl*, *acl*, *advcl*, and *cc*) can be done on the basis of UD annotation alone. We return to these in the section comparing the two conversion methods.

Technically, EUD differs from UD in that it contains null nodes. Furthermore, the annotation graph is no longer guaranteed to be a tree (e.g. the treatment of control means that an element can be the dependent of two predicates), may contain cycles (a consequence of the analysis of relative clauses), and may lack a root node.

It is an open question whether adding EUD to a corpus that already has been annotated with UD can be done automatically, using heuristics for those cases where information is lacking, or whether it requires additional supervision, either in the form of manual correction or information obtained from an underlying treebank that already has annotation in place for elided predicates, shared conjuncts, and raising and control. Here we compare these two possibilities for the Dutch treebanks by comparing the outcome of a rule-based conversion script that takes the underlying treebank annotation as input, with the method of Schuster et al. (2018). The latter paper presents an automatic method for converting from UD to EUD that was tested on English, Finnish, and Swedish. It can be adapted to other languages by providing word embeddings for that language and a list of relative pronoun forms. We ran it with these modifications on the Dutch UD treebanks.

2 Conversion to EUD from an Underlying Treebank

In this section we describe our method for adding enhanced dependency annotation to a corpus for which manually verified language specific syntactic annotation is already in place.

There are currently two Dutch UD corpora. They are based on manually verified treebanks that both follow the guidelines of the Lassy treebank project (van Noord et al., 2013). The Lassy annotation guidelines combine elements from phrase structure treebanks (such as phrasal nodes and use of co-indexed nodes for encoding the syntactic role of fronted WH-constituents, relative pronouns, and ellided phrases) with elements from dependency treebanks (such as dependency labels, discontinuous constituents and crossing branches), similar to the Tiger (Brants et al., 2002) and Negra (Skut et al., 1998) corpora for German.

As of release v2.2 of the Universal Dependencies corpora, (portions of) both Dutch corpora have been converted to UD using the same conversion script.[4] An example of the conversion process is given in Figure 1. For each phrasal node in the underlying annotation, the script first identifies

[4]Available at `https://github.com/gossebouma/lassy2ud`

the head daughter according to UD (indicated by a boxed node in the diagram). In most cases, this is a leaf node labeled with the underlying dependency relation hd. In some cases, such as PPs, however, the UD guidelines and the underlying annotation diverge and (the head of) a non-head daughter is selected as the UD head. Next, labeled edges between non-null leaf nodes are added. For a head node H and a sister non-head node D, an edge $H \xrightarrow{label} D$ is added if H and D are lexical nodes. If H or D is phrasal, the dependency holds between the lexical heads of H and/or D. The label is determined by a mapping from the underlying dependency label of D to the corresponding UD label (where in some cases, the mapping depends on other aspects of the syntactic context). Note that null nodes in the underlying annotation are not annotated in the conversion to UD. The conversion process is described in detail in Bouma and van Noord (2017).

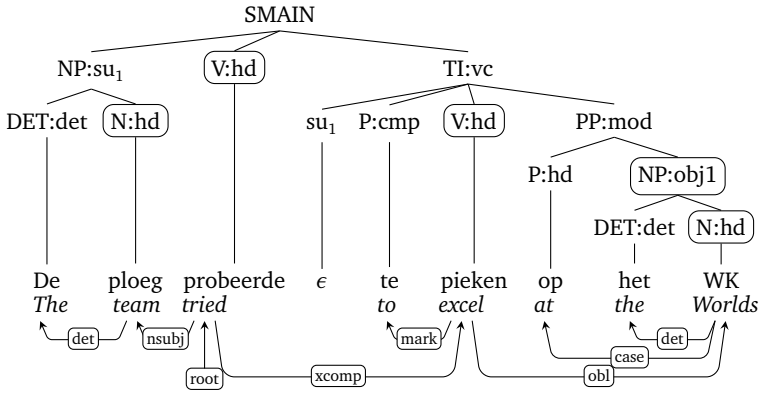


Figure 1: Phrasal annotation and the induced dependency annotation for *de ploeg probeerde te pieken op het WK* (*the team tried to excel at the world championships*).

We add EUD as follows: first, we recover elided predicates in the Alpino dependency tree. Next, we run the standard Alpino to UD conversion script on the expanded tree while ensuring that dependency labels for relations *obl*, *nmod*, *acl*, *advcl*, and *conj* are expanded so as to include the lemma form of the relevant *case, mark* or *cc* dependent or other subtyping (e.g. relatives are labeled *acl:relcl*). Finally, we add dependency relations resulting from the propagation of conjuncts, the modified analysis of relative clauses, and the explicit annotation of control relations. Below, we describe the steps that require access to aspects of the underlying annotation, i.e. recovery of elided predicates, propagation of conjuncts, and adding control relations.

2.1 Recovering elided predicates

The standard UD annotation of gapping and ellipsis (Schuster et al., 2017) involves promoting the highest ranked dependent to head and attaching any remaining dependents to this pseudo head using the *orphan* relation, as illustrated in Figure 2 (top annotation). In EUD, elided predicates are reconstructed by adding additional lexical elements to the input string that are copies of the preceding or following head node to which they correspond. The enhanced annotation can refer to these inserted lexical elements as well, thus making the process that promotes dependents to heads and the *orphan* dependency label superfluous (Figure 2, bottom annotation).

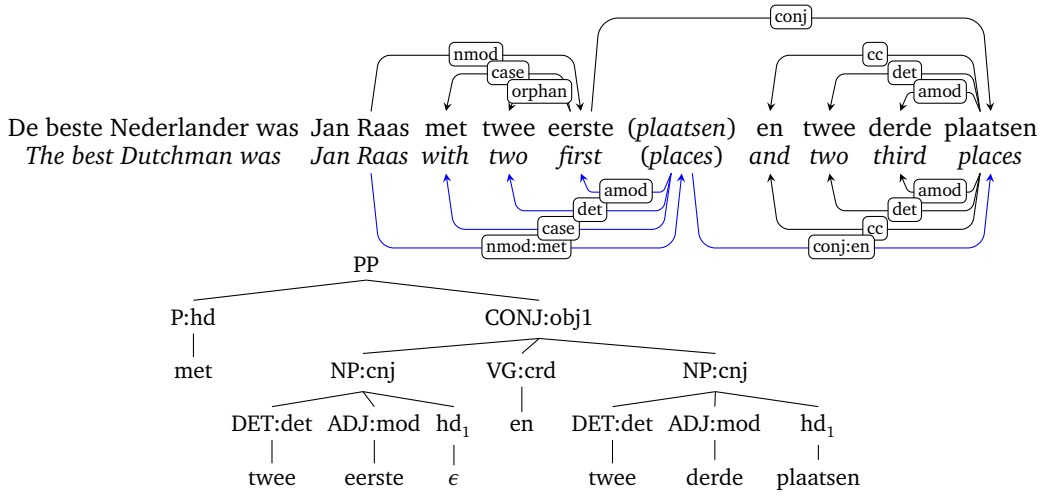


Figure 2: UD (above) and EUD (below) annotation and Alpino dependency tree for the sentence *The best Dutchman was Jan Raas with two first and two third places* containing an NP coordination with an elided nominal predicate in the first conjunct. Co-indexing is shown using subscripts.

As the underlying (Alpino) annotation contains null nodes for elided elements in ellipsis constructions, the recovery of elided predicates can be done as part of the automatic conversion by replacing the relevant null nodes (that is, a co-indexed empty node that is a head according to UD) with a node that is a copy of the head node with which it is co-indexed. With these recovered elements in place, dependency labels can be re-computed and the need to promote non-heads to heads disappears.

2.2 Propagation of Conjuncts

When the subject or object is a conjoined phrase, an additional dependency relation labeled as *subj* or OBJ from (verbal) predicate to the non-head *conj* dependent is added. This addition does not require access to underlying annotation. The guidelines for EUD also state that subjects, objects, and other complements that occur as dependents of a conjoined predicate are attached to both predicates, as illustrated in Figure 3. In the underlying annotation, cases like these (i.e. conjoined predicates which share one or more dependents) are easily recognized as the shared dependent present in one of the conjuncts is co-indexed with an empty node in the other conjunct.

2.3 Raising and control

The guidelines state that for embedded clauses in raising and control contexts, an additional subject relation should be added from the embedded verb to the controlling subject or object in the matrix clause (Figure 4). In the underlying annotation for such cases, the embedded clause contains an empty subject node, co-indexed with a controlling subject, object, or indirect object in the matrix clause. In the conversion to EUD, a dependency relation is added from the head of the embedded clause to (the head of) the controlling NP.

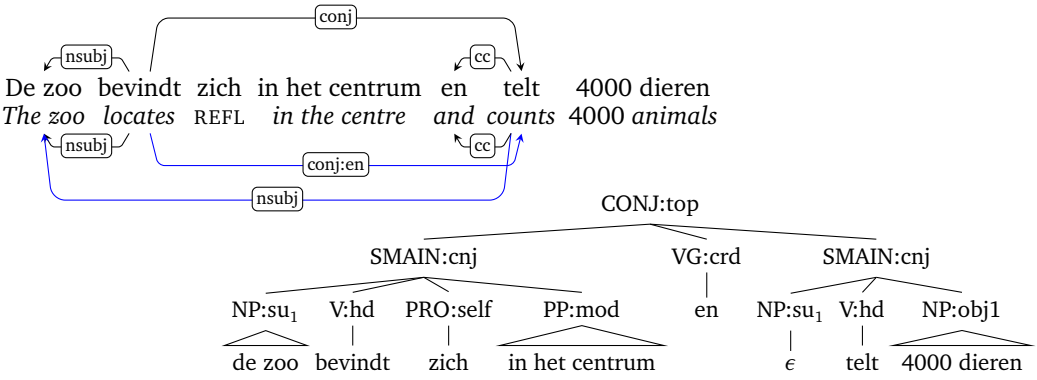


Figure 3: UD (above) and EUD (below) annotation and Alpino dependency tree for a sentence coordination with an elided subject in the second conjunct for the sentence *The zoo is located in the center and has 4,000 animals*.

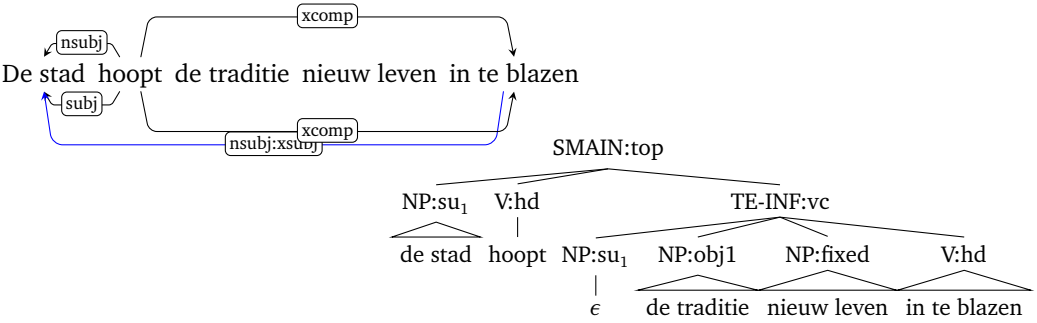


Figure 4: UD (above) and EUD (below) annotation and Alpino dependency tree for the sentence *The city hopes to bring the tradition to life again*. Note that we follow the convention of Schuster et al. (2018) of adding an extension to the *nsubj* label indicating its status.

The conversion takes into account the fact that the controller may itself be co-indexed, as illustrated in Figure 5, where the subject of a passive auxiliary is co-indexed with an elided subject in the second conjunct, which in turn controls a subject in the embedded clause.

2.4 Relative Clauses

In relative clauses, a direct dependency is added from the head of the relative to the nominal antecedent. The dependency is labeled with the dependency relation corresponding to that of its relative pronoun in the standard UD annotation. The dependency to the relative pronoun itself is relabeled as *ref* (Figure 6). Note that the EUD annotation contains a cyclic dependency ($snelweg \xrightarrow{acl:relcl} omcircelt \xrightarrow{nsubj:relsubj} snelweg$).

When the relative pronoun is missing, it seems the correct label for the dependency relation from the head of the relative clause to the antecedent noun requires access to additional information (such as an empty node in the underlying annotation). Unlike English, Dutch does not have

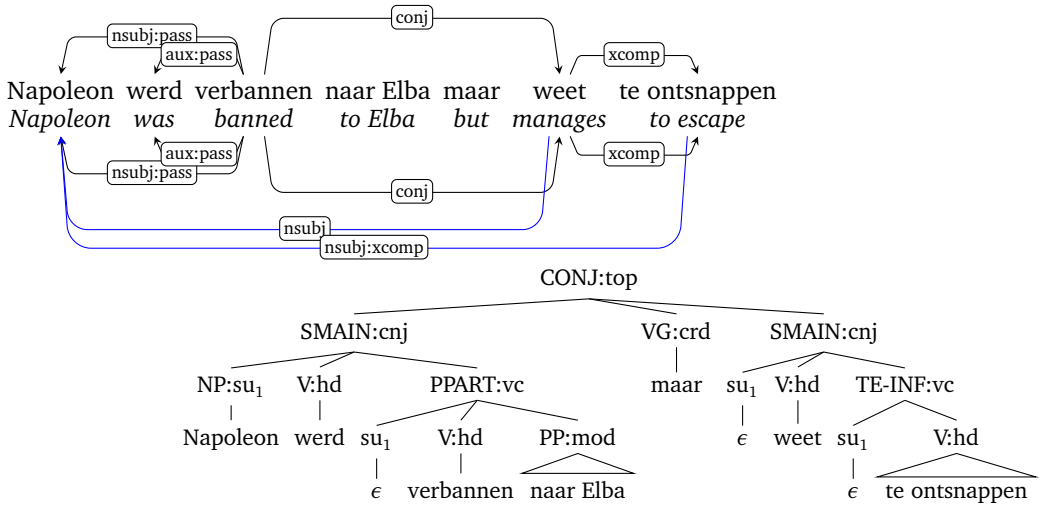


Figure 5: UD and EUD (below) annotation and underlying annotation for the sentence *Napoleon was banned to Elba but manages to escape.*

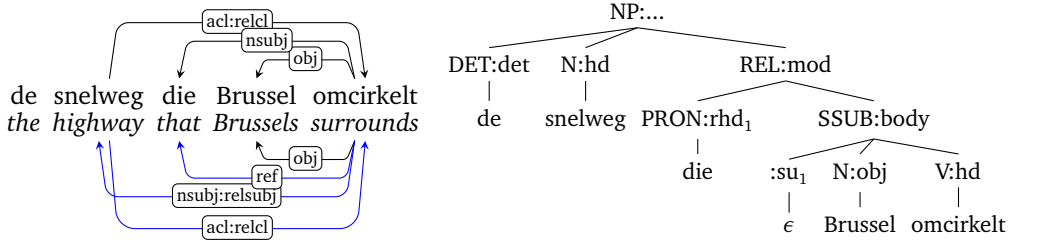


Figure 6: UD and EUD annotation and underlying annotation (right) for the NP *the highway that surrounds Brussels.*

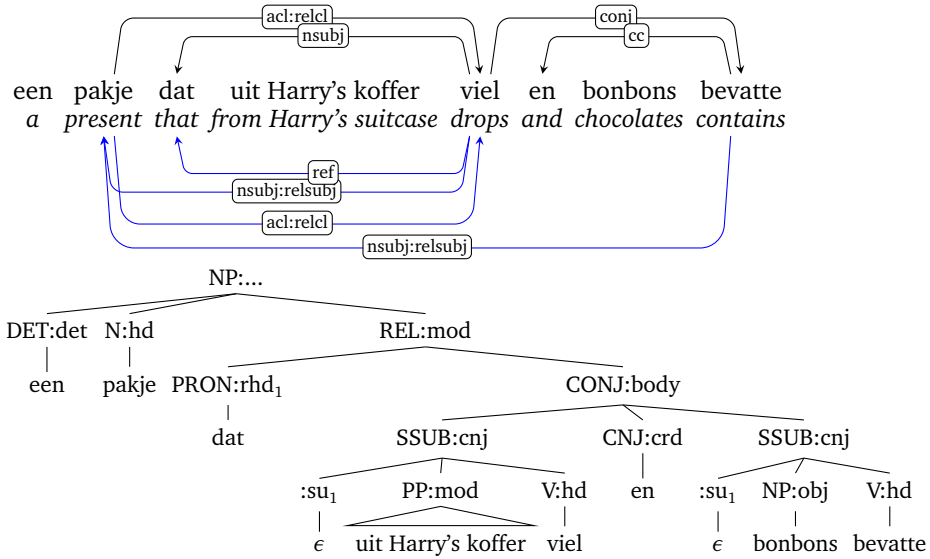


Figure 7: UD and EUD (top) annotation and underlying annotation (bottom) for the NP *a present that drops from Harry's suitcase and contains chocolates.*

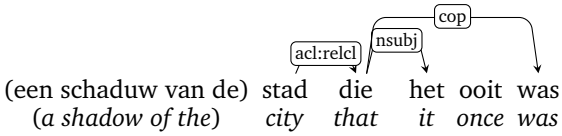


Figure 8: UD annotation for the NP *a shadow of the city that it once was*, where the relative pronoun heads the relative clause.

bare relatives and thus, at first sight, this problem does not arise. It turns out, however, that coordinated relative clauses, where the relative pronoun heads a conjunction and is "*extracted across-the-board*", do give rise to a similar problem. In these cases, illustrated in Figure 7, the grammatical role of the antecedent noun in the second conjunct is not explicitly marked in UD, as the relative pronoun is a dependent of the head of the first conjunct only. All instances of this phenomenon in the Dutch treebanks exhibit strict parallelism (i.e. the relative pronoun fulfills the same grammatical role in both conjuncts) but whether this is a requirement has been a matter of some debate (Williams, 1978; Hartmann et al., 2016).

Another complication arises in cases where the relative pronoun is a predicate in a copula construction. UD considers the predicate to be the head in copula constructions, and thus the relative pronoun is the syntactic head of the relative clause in cases such as Figure 8. The relative pronoun thus is an *acl:relcl* dependent of the antecedent noun. The EUD guidelines for relative clauses potentially introduce a reflexive dependency in such cases (from antecedent noun to antecedent noun, replacing the UD dependency pointing to the relative pronoun). As this is not allowed (and its interpretation would be unclear), the UD annotation is preserved in EUD in such cases.[5]

[5]The current annotation guidelines acknowledge this problem.

	UD deps	EUD deps	EUD=UD	elided
AE	75,100	78,203	63,491	84
SE	75,100	78,200	63,398	38

Modified deps	deps	%
Agreement	13,463	85.7
Disagreement	2,251	14.3
Total	15,714	100.0

relation	AE	SE	relation	AE	SE
acl:relcl	501	533	nsubj	3,645	3,761
advcl:om+acl:om	89	19	nsubj:pass	721	742
advcl:te	7	106	nsubj:pass:relsubj	53	64
advmod:*	2,492	2,538	nsubj:relsubj	276	295
amod:*	4,513	4,376	nsubj:xsubj	194	403
aux:pass	762	753	obl:*	4,680	4,655
det	7,944	7,815	ref	484	395
nmod	6,272	6,164	root	5,788	5,733

Table 1: Statistics for the effect of going from UD to EUD. The top-right table shows (dis-) agreement between AE and SE only for those cases where the EUD of either AE or SE is not equal to UD.

3 Comparison

We ran both the rule-based method outlined above which takes the underlying language specific (Alpino) annotation as starting point (referred to as AE in this section) and the heuristic method of Schuster et al. (2018) (SE) on the training section of the UD Dutch LassySmall corpus.[6]

Table 1 (top-left) shows that the EUD has approx. 4% more relations than UD for both AE and SE, but also that the relations present in EUD are identical to UD in only approx. 84% of the cases. The number of newly introduced elided nodes differs significantly between the two methods. The top-right table gives statistics for the level of agreement between AE and SE, for those cases where EUD $\neq$ UD in either AE or SE.

The bottom table in Table 1 gives an overview of dependency relations whose frequency differs most strongly between AE and SE. The differences can be attributed to the following phenomena:

- **Propagation of conjuncts**: if the underlying annotation indicates that modifiers and function words should be spread across conjuncts, AE applies the rule for propagation of conjuncts, whereas SE does not. This explains the higher numbers for *amod,det, nmod* and *obl* in AE. It accounts for phrases such as *verlaten straten en ruines (deserted streets and ruins), talrijke persoonlijke documenten en brieven (numerous personal documents and letters), de herkomst en geschiedenis van woorden (the origin and history of words)* and *invoer of uitvoer van buitenlandse deviesen (import and export of foreign currencies)*.

- **Lack of propagation of conjuncts**: If a predicate in a copula construction is conjoined (as in *Mandeville was arts en filosoof (Mandeville was writer and philosopher)*), no co-indexed

[6]There was one sentence, wiki-8628.p.21.s.3, for which SE did not produce a result. It contains a coordinated relative clause, with a severely elliptical second conjunct (where in the underlying treebank, the relative pronoun, passive auxiliary, and passive participle are elided). It is excluded in the comparison below.

empty subject is present in the underlying annotation (as in that annotation, the copula is the head and not the predicate). As a consequence, no propagation of conjuncts takes place in AE, whereas these cases are covered by SE. This explains the higher number of *nsubj* in SE. Also, SE spreads conjoined relative clauses where AE does not. This explains why SE contains more *acl:relcl* and *nsubj:relsubj* dependents.

- **Adding control relations**: SE adds a control relation to all *xcomp* dependents. This includes non-verbal cases such as *het stripboek was bedoeld als reclame (the comic was intended as advertisement), de partij kwam in opspraak (the party became controversial), door een campagne te voeren met als thema ....(by running a campaign with as theme ....)*. As these have no co-indexed empty subject in the underlying annotation, AE does not add a control relation in these cases and thus it has significantly fewer *nsubj:xcomp* dependents. On the other hand, AE correctly identifies the indirect object as controller in cases like *de priester gaf Johannes te drinken (the priest gave Johannes to drink)*, where *Johannes* is a *iobj* and *te drinken* a non-finite verbal *xcomp*.

- **Disagreements about lexical extensions:** The dependencies *acl* and *advcl* are extended with the lemma of their *case* or *mark* dependent. However, *advcl* and *acl* clauses can contain both, as in *een zondag om nooit te vergeten (a sunday to never forget)*. AE adds the *case* lemma *om* in these cases, where SE adds the marker *te*. This explains the almost complementary distribution of *acl:om* + *advcl:om* and *acl:te* + *advcl:te*. A similar issue arises with coordinations containing two *cc* elements, such as *zowel op straat als in woningen (both on the street and in houses)*, where AE adds the lemma of the first *cc* to the *conj* dependency and SE the lemma of the second. Finally, some prepositional phrases consist of a preposition as well as a postpositional particle (*op twee na (except two)*). Again, the two methods give different results for the lexical extension.

- **Graphs without root:** A consequence of the analysis of relatives is that some sentences do not have a root element. I.e. in *Kuifje is een reporter die in Brussel woont (Tintin is a reporter who lives in Brussels)*, *reporter* is the root in UD, but it is a dependent of *lives* in EUD, thus loosing its root status according to SE but not according to AE. The documentation for UD simply states that the *root* dependency points to the root of the sentence, while EUD does not raise the issue whether a EUD graph can be without root or even have multiple roots, thus leaving open the question what exactly defines a root in a potentially cyclic graph (i.e. is it the element that is not a dependent of anything else or the element(s) from which all other elements can be reached?).

- **Position of elided predicates:** A last systematic source of conflict between AE and SE, not immediately reflected by the statistics in Table 1, is disagreement about the position of an elided item. Apart from the fact that AE inserts more elided elements than SE, it is also the case that AE and SE often disagree about the exact position of the elided material. Remember that in the CONLL-U format[7], elided nodes are entered as additional lines in the annotation, indexed with a subscripted (N.1 *etc.*) node. The annotation guidelines do not specify explicitly where to place this element in the string. A human annotator would most likely place the elided predicate at the position which results in a grammatical sentence. The AE script, however, recovers elided predicates by spotting index nodes without lexical content that lack attributes indicating their string position. The internal XML-format of the

[7] `universaldependencies.org/format.html`

Alpino annotation trees employs a normal form where, by default, empty index elements are leftmost daughters of the phrasal node by which they are dominated. The AE script therefore inserts recovered elided elements in front of the words making up the phrase in the corresponding CONLL-U format. SE usually opts for a more natural word order. Note that as a consequence, there not only is a conflict regarding the elided predicate itself, but also all dependents of this element will have conflicting EUD annotations. It is not clear to us however whether the AE strategy violates the guidelines or not.

Most of the conflicts between the two methods can be resolved by making the guidelines more explicit (i.e. by listing which dependents should be included in spreading of conjuncts, by deciding whether every *xcomp* should have a subject, by providing a definition of *root*, and by specifying where elided elements should be placed). We do not think that there are decisions that would be hard to implement by either the rule-based or the heuristic approach. The rule-based approach can obviously be modified so as to include spreading of conjoined predicative heads and relatives, to filter *root* dependency labels of nodes that have another dependency label as well, and to add a controller to non-verbal XCOMP's. The most challenging problem might be the adequate placement of elided predicates, but then, it should be noted that it is unclear to what extent this is relevant for downstream (semantic) applications. The heuristic strategy might seem to be hard pressed to predict propagation of conjuncts that are modifiers, such as *amod*, *nmod*, and *obl*. A brief corpus investigation suggests however, that, for instance, spreading of *amod* dependents occurs almost always in those cases where the *amod* is part of the first conjunct, and the second conjunct is not introduced by a determiner and almost never contains an *amod* itself. Spreading of *obl* occurs almost always in conjoined finite clauses where the phrase headed by *obl* occurs as first (fronted) constituent and the second conjunct is verb-initial (i.e. does not contain a fronted element).

4 Conclusion

In this paper, we have outlined a rule-based approach for converting a treebank with rich but language-specific dependency annotation to EUD and compared this with a method that converts from UD to EUD. Our hypothesis was that the richer annotation in the underlying treebank would help to produce more accurate annotation than the heuristic method that uses information from UD only. Comparing the output of both conversion scripts on the training section of UD Dutch LassySmall revealed that there are systematic differences between the two. These appear to be mostly due to insufficient detail in the annotation guidelines for EUD. Resolving these would probably reduce the number of conflicts in the output of the two conversion methods to a very small number, suggesting that both the AE and the SE method are able to produce accurate EUD annotation.

Some issues regarding the guidelines may require considering once more which problem EUD is supposed to solve exactly. Candito et al. (2017) cover a somewhat different set of syntactic phenomena that might also be incorporated in the enhanced layer. They propose to specify control also for certain adjectives and non-finite verbal modifiers of nouns and furthermore argue for a canonical representation that undoes the effect of diathesis alternations. From a semantic point of view, these are valuable enhancements, and it seems to fit in the general philosophy of UD that says that comparable sentences should receive comparable annotation across languages. Schuster and Manning (2016), on the other hand, discuss certain enhancements of basic UD such as a reanalysis of partitives, light noun constructions, and phrasal prepositions, that are

specific to English. More generally, one can imagine that each language specifies what counts as a partitive, light verb, phrasal preposition, etc. (not unlike specifying what counts as an auxiliary in the current UD annotation scheme), and that the annotation or conversion can proceed automatically and uniformly across languages on the basis of such lists. The utility of such enhancements is demonstrated for instance in Bar-Haim et al. (2007), which contains a series of normalization and inference rules over syntactic (dependency) trees for recognizing textual entailment. At the same time, it seems that EUD ideally should also function as a uniform syntactic representation that can be used to derive full semantic interpretations of sentences without requiring language-specific lexica or rules. It remains to be seen whether the complex graphs that result from adding all conceivable enhancements are indeed suitable for such systems as well.

Acknowledgments

This research was carried out while the author was visiting scientist at the Center for Advanced Study at the Norwegian Academy of Science and Letters. It has benefitted from presentations at the Center as well as from feedback by the anonymous reviewers for this conference.

References

Bar-Haim, R., Dagan, I., Greental, I., Szpektor, I., and Friedman, M. (2007). Semantic inference at the lexical-syntactic level for textual entailment recognition. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 131–136. Association for Computational Linguistics.

Bouma, G. and van Noord, G. (2017). Increasing return on annotation investment: the automatic construction of a Universal Dependency treebank for Dutch. In Nivre, J. and de Marneffe, M.-C., editors, *NoDaLiDa workshop on Universal Dependencies*, Gothenburg.

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER treebank. In *Proceedings of the workshop on Treebanks and Linguistic Theories*, volume 168.

Candito, M., Guillaume, B., Perrier, G., and Seddah, D. (2017). Enhanced UD dependencies with neutralized diathesis alternation. In *Depling 2017-Fourth International Conference on Dependency Linguistics*.

Futrell, R., Mahowald, K., and Gibson, E. (2015). Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Gotham, M. and Haug, D. (2018). Glue semantics for universal dependencies. In *Proceedings of the 23rd international Lexical-Functional Grammar Conference*, Vienna.

Hartmann, J., Konietzko, A., and Salzmann, M. (2016). On the limits of non-parallelism in ATB movement: Experimental evidence for strict syntactic identity. *Quantitative Approaches to Grammar and Grammatical Change: Perspectives from Germanic*, 290:51.

Lipenkova, J. and Souček, M. (2014). Converting Russian dependency treebank to Stanford typed dependencies representation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 143–147.

Przepiórkowski, A. and Patejuk, A. (2018). From LFG to enhanced universal dependencies. In *Proceedings of the 23rd International LFG conference*, Vienna.

Pyysalo, S., Kanerva, J., Missilä, A., Laippala, V., and Ginter, F. (2015). Universal dependencies for Finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (Nodalida 2015)*, pages 163–172.

Reddy, S., Täckström, O., Petrov, S., Steedman, M., and Lapata, M. (2017). Universal semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 89–101.

Schuster, S., Lamm, M., and Manning, C. D. (2017). Gapping constructions in universal dependencies v2. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 123–132.

Schuster, S. and Manning, C. D. (2016). Enhanced English universal dependencies: An improved representation for natural language understanding tasks. In *Proceedings of LREC*.

Schuster, S., Nivre, J., and Manning, C. D. (2018). Sentences with gapping: Parsing and reconstructing elided predicates. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2018)*.

Shwartz, V., Goldberg, Y., and Dagan, I. (2016). Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the ACL 2016*, pages 2389–2399, Berlin.

Skut, W., Brants, T., Krenn, B., and Uszkoreit, H. (1998). A linguistically interpreted corpus of German newspaper text. *arXiv preprint cmp-lg/9807008*.

van Noord, G., Bouma, G., van Eynde, F., de Kok, D., van der Linde, J., Schuurman, I., Sang, E. T. K., and Vandeghinste, V. (2013). Large scale syntactic annotation of written Dutch: Lassy. In Spyns, P. and Odijk, J., editors, *Essential Speech and Language Technology for Dutch: the STEVIN Programme*, pages 147–164. Springer.

Vulić, I. (2017). Cross-lingual syntactically informed distributed word representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 408–414.

Wang, Y. and Liu, H. (2017). The effects of genre on dependency distance and dependency direction. *Language Sciences*, 59:135 – 147.

Williams, E. (1978). Across-the-board rule application. *Linguistic Inquiry*, 9(1):31–43.

Zeman, D., Popel, M., Straka, M., Hajic, J., Nivre, J., Ginter, F., Luotolahti, J., Pyysalo, S., Petrov, S., Potthast, M., Tyers, F., Badmaeva, E., Gokirmak, M., Nedoluzhko, A., Cinkova, S., Hajic jr., J., Hlavacova, J., Kettnerová, V., Uresova, Z., Kanerva, J., Ojala, S., Missilä, A., Manning, C. D., Schuster, S., Reddy, S., Taji, D., Habash, N., Leung, H., de Marneffe, M.-C., Sanguinetti, M., Simi, M., Kanayama, H., dePaiva, V., Droganova, K., Martínez Alonso, H., Çöltekin, c., Sulubacak, U., Uszkoreit, H., Macketanz, V., Burchardt, A., Harris, K., Marheinecke, K., Rehm, G., Kayadelen, T., Attia, M., Elkahky, A., Yu, Z., Pitler, E., Lertpradit, S., Mandl, M., Kirchner, J., Alcalde, H. F., Strnadová, J., Banerjee, E., Manurung, R., Stella, A., Shimada, A., Kwak, S., Mendonca, G., Lando, T., Nitisaroj, R., and Li, J. (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.

The Potsdam Commentary Corpus 2.1 in ANNIS3

Peter Bourgonje, Manfred Stede

Applied Computational Linguistics
University of Potsdam, Germany

bourgonje|stede@uni-potsdam.de

ABSTRACT

We present a new version of the Potsdam Commentary Corpus; a German corpus of news commentary articles annotated on several different layers. This new release includes additional annotation layers for dependency trees and information-structural aboutness topics as well as some bug fixes. In addition to discussing the additional layers, we demonstrate the added value of loading the corpus in ANNIS3, a tool to merge different annotation layers on the same corpus and allow for queries combining information from different annotation layers. Using several cross-layer example queries we demonstrate its suitability to corpus analysis for various different areas.

KEYWORDS: treebanks, information structure, cross-layer analysis.

1 Introduction

The Potsdam Commentary Corpus (PCC) was introduced by Stede (2004) as a collection of 176 newspaper editorials (comprising over 34k words in over 2100 sentences) from a German regional newspaper, which had been manually annotated on different layers: sentence syntax, coreference, and rhetorical structure. More recently, an updated version (PCC 2.0) was presented by Stede and Neumann (2014); besides major revisions on the rhetorical structure and coreference layers, it included an additional layer of connectives and their arguments, similar in spirit to the annotations in the Penn Discourse Treebank (Prasad et al., 2008).

In this paper, we present the new release PCC 2.1, which offers three new features:

- A new layer of manually-annotated information-structural aboutness topics (cf. (Stede and Mamprin, 2016))
- A new layer of automatically-produced dependency parses
- Availability of the 'manual' layers in the ANNIS3 linguistic database (Krause and Zeldes, 2016)

The integration in ANNIS3 allows for qualitative studies on the interactions among the various layers of annotation, and visualization of search results.

In the following, we provide background information on the corpus and the ANNIS3 system, briefly discuss the technical conversion from annotation tool files to ANNIS3, and then discuss some sample queries in order to illustrate the potential for cross-layer analyses. The paper concludes with an outlook on our next steps.

2 Corpus and Database

2.1 PCC

The PCC was deliberately built as a genre-specific corpus, mainly intended for studying the textual means for expressing opinion and argumentation in the German language. In order to support these high-level goals, a number of lower-level (i.e., closer to the linguistic surface) phenomena have been annotated manually, so that a gold standard is available for evaluating automatic experiments, and also for manually studying the interactions of phenomena. Thus, sentence syntax was annotated (in the early 00's) by the Potsdam team of the TIGER project (Brants et al., 2002), and (nominal) coreference annotation built on the proposals of the PoCoS annotation scheme suggested by Krasavina and Chiarcos (2007). In recent years, all annotations (except for syntax) have been revisited and sometimes changed, so that they now reflect the annotation guidelines, which are collectively available in the volume (Stede, 2016).

Text structure, in the spirit of Rhetorical Structure Theory (Mann and Thompson, 1988), has been at the center of interest in the PCC from the beginning. The main postulate of RST is that a text can be segmented into so-called *elementary discourse units* (sentences, certain types of clauses), and that *coherence relations* connect adjacent text spans – which can be either elementary units, or recursively-built larger spans. A set of some 20 relations is suggested, with various causal, temporal, contrastive and additive relations among them. For the majority of the relations, the connected units have different statuses of prominence: A *nucleus* is more important for the author's purposes, and the *satellite* unit supports that purpose but is overall less important.

(Matthiessen and Thompson (1988) discuss the relationship between the nucleus/satellite dichotomy and syntactic subordination.) By this analysis, a text is ultimately represented as a single tree structure that spans the entire text. See (Mann and Thompson, 1988) for a full explanation. The RST annotations in the PCC served as train/test data for the first RST parser developed for German (Reitter, 2003). Until today, PCC is the largest German-language RST resource. The ANNIS team built a specific visualization module for these discourse trees, which reflect the recursive application of coherence relations.

A similar layer of annotations holds the lexical connectives (coordinating and subordinating conjunctions, various adverbials, a few prepositions)[1] and their arguments. However, following the spirit of the Penn Discourse Treebank (Prasad et al., 2008), they do not combine into any description of a text structure; instead, they are annotated individually and without taking other connective/argument constellations into consideration. This, in turn, allows for posthoc studying the correlations between connectives/arguments on the one hand, and rhetorical structure on the other. Notice that, in contrast to the PDTB, *implicit* coherence relations have not been annotated in the connective layer of the PCC, as this would effectively duplicate parts of the RST annotation task.

The latest layer of annotation concerns *aboutness topics*, with annotation guidelines being inspired by the characterization given by Jacobs (2001). In line with earlier work, we regard the aboutness topic as the syntactic constituent referring to the entity about which information is communicated by the central predication of the sentence. According to Jacobs, a 'prototypical' aboutness topic fulfils three criteria:

- Informational separation: The topic precedes the predication, and semantic processing of the sentence is correspondingly done in two subsequent steps.
- Predication: The topic fills a position in the valency frame of the predicate. (It is not an adjunct.)
- Addressation: The topic refers to an entity that serves as the 'address' for storing information in the common ground of speaker and hearer.

Reliably identifying topics in authentic text as opposed to single-sentence "laboratory examples" can be difficult, though (Cook and Bildhauer, 2013). Therefore, our guidelines had to make a range of additional commitments, concerning primarily the partitioning of complex sentences into clauses that should (or should not) receive a topic annotation, and the handling of incomplete sentences, i.e., fragmentary material. The annotation effort is documented in (Stede and Mamprin, 2016); annotator agreement is κ 0.6 for the theticity question (should a discourse unit be assigned a topic or not), and κ 0.71 for selecting the aboutness topic.

Finally, with release 2.1 we now began to also include automatic annotations for the purposes of providing training/test data for further automatic analysis tasks. Specifically, we added dependency parses produced by the ParZu system (Sennrich et al., 2009).[2]

In addition to these extra layers, the 2.1 release includes several minor bug fixes on different layers of the manual annotation.

[1] For discussion, see (Danlos et al., 2018), and for lists of connectives in various languages `http://www.connective-lex.info`

[2] Thanks to Don Tuggener for providing us with this data.

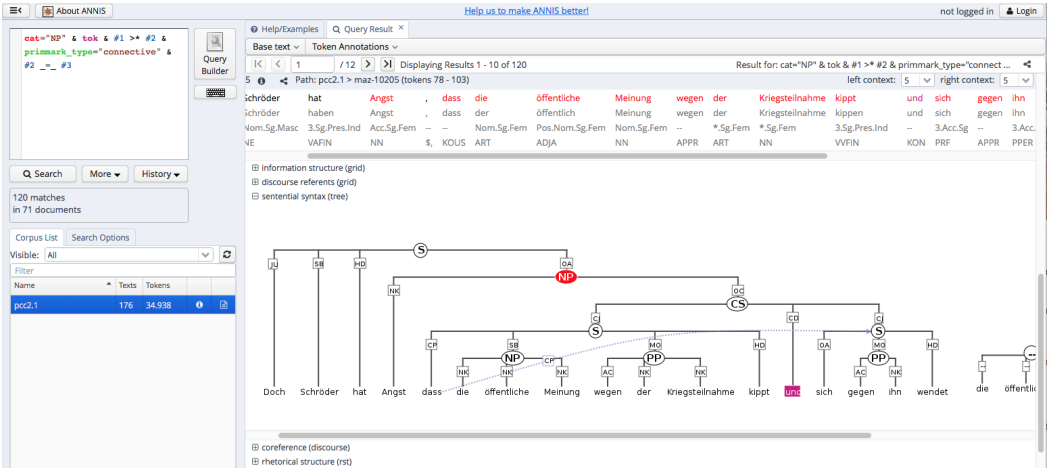


Figure 1: Screenshot of ANNIS3 Database

2.2 ANNIS

The first version of the ANNIS database (Dipper et al., 2004) was developed at Potsdam University in order to support research on information structure by providing an infrastructure for merging (manual or automatic) annotations of the same data into a single representation, and allowing queries across the various layers. Subsequently, further developments, including a complete redesign, was done largely at Humboldt University Berlin, and the scope of applications widened considerably. The latest release ANNIS3[3] offers, *inter alia*, the inclusion of audio/video data and support for many languages and character sets, and it contains modules for mapping the output of many manual annotation tools to the native database format. The system is available both as a server version and as a standalone app that can be run on laptops. See (Krause and Zeldes, 2016) for an overview.

The different visualization modules of ANNIS3 provide

- graphic representations of constituency trees and dependency trees for sentence syntax;
- graphic representations of discourse trees (inspired by the format used in RSTTool[4]);
- highlighting of co-referring elements in a text view;
- presentation of other kinds of span annotations in a "grid" view (inspired by the format used in Exmaralda[5]).

In addition, parallel aligned data can be represented, and inspected with the help of co-colouring the aligned elements.

The Annis Query Language (AQL) allows to search for patterns in the primary text (e.g., via regular expresions) and in the annotations. As for the latter, one can look for simple labels

[3] http://corpus-tools.org/annis/
[4] http://www.wagsoft.com/RSTTool/
[5] http://exmaralda.org/de/partitur-editor-de/

associated with annotation objects, direct and indirect dominance relations in trees, and for so-called 'pointing relations' as they can be used, for example, for coreference annotation. For illustration, Figure 1 shows a screenshot with a query, set of hits, and a syntax tree selected for visualization. This is one example of a query spanning across multiple layers of annotations and hence combining different linguistic phenomena; it will be discussed in Section 4.

3 Tranferring annotation files to ANNIS3

ANNIS3 comes with separate tools for importing, merging and exporting data sets; Salt[6] and Pepper[7]. Pepper, the tool responsible for importing data from several formats, in turn comes with a collection of existing importing modules, supporting many popular annotation file formats, including CoNLL, MMAX2, PTB, Tiger2 and many more[8]. For all but one annotation layer of PCC 2.1, an importer was readily available to merge the annotations of different layers into the format that is directly read by ANNIS3; the aboutness topics were annotated in the EXMARaLDA format; the sentential syntax (constituency trees) were available in the Tiger2 format; the discourse referents (coreference layer) were in MMAX2 format, and the RST trees in rs3 format (the native format of the RST Tool). The exception for which no importer was available, was the annotation layer for discourse relations, which were in a custom inline XML format, as produced by the annotation tool Connanno[9]. Although Pepper comes with documentation on how to extend the toolkit with additional importing modules, in this case it was simpler to convert this custom XML format to a supported one (MMAX2) and then to import. From there on, all annotation layers are easily merged, with the important prerequisite being that tokenisation is consistent across all annotation layers. After importing and merging all layers, the result is exported to a format that is finally loaded into the PostgreSQL database, which ANNIS3 is reading from and then enables querying and visualisation.

4 Querying the multi-layer PCC

In this section, we provide some examples to illustrate the potential of cross-layer queries in the PCC. Notice that the various layers have all been annotated independently of each other (and at quite different points in time), so that exploring correlations does not simply reproduce the complete set of choices made in one shot by a single annotator.

Interesting uses of connectives. The German word "da" has a number of readings, one of them being that of a subordinate conjuntion, where it routinely plays the role of a causal (or argumentative) connective. To begin, the simple query `"da" | "Da"` yields 56 hits, and 8 of these are labelled as connectives. Zooming in on subordinate clauses following the main clause (i.e., on the non-capitalied "da"), we get 4 connectives. Of these, one turns out to be somewhat non-standard because the syntax layer does not tag it as subordinate conjunction. The corresponding query is:

```
tok = "da" & primmark_type = "connective" & pos != "KOUS"
& #1 _=_ #2 & #2 _=_ #3
```

And the hit is *jetzt, da das gesamte Paket überschaubar wird* ('now, as the complete package becomes visible'). And indeed, the annotated sense of this connective is not, as usual, causal but temporal.

[6] `http://corpus-tools.org/salt/`
[7] `http://corpus-tools.org/pepper/`
[8] See `http://corpus-tools.org/pepper/knownModules.html` for the exhaustive list.
[9] `http://angcl.ling.uni-potsdam.de/resources/connanno.html`

Embedded discourse units. One point of debate in discourse structure theories (e.g., (Hoek et al., 2018)) is the proper handling of 'elemenary discourse units' that are syntactically embedded. Using the AQL operator for transitive domination in trees, we can for instance look for connectives that are embedded in an NP:

```
cat="NP" & tok & #1 >* #2 & primmark_type="connective" & #2 _=_ #3
```

This query, which is shown above in Figure 1, yields 120 hits. The screenshot shows the relatively common case of a noun modified by a complement clause ('the fear that public mood will change because of entering the war *and* turn against him'). With queries like this, such "isolated" connective annotations can be set in correspondence with the syntactic configurations, and then the consequences for an approach to representing complete discourse structures, for example in RST, can be pondered.

Aboutness topic and grammatical role. The "typical" aboutness topic in a sentence is the grammatical subject (cf., e,g., (Jacobs, 2001)). Using ANNIS queries, we can compute the set of topics that are *not* subjects and then determine the reasons. The PCC contains 1.417 topics, 1.184 of which have a grammatical role annotation (the others merely overlap with a syntactic unit with a role annotation). Of these, in turn, 347 have a role other than subject, which amounts to 24% of all aboutness topics. A qualitative investigation of 90 instances shows that the largest group is due to the subjects being impersonal (e.g., *man*/'one') or expletive pronouns, which prompted annotators to associate the topic label with a different constituent. In the group of more interesting cases, we find subjects that are non-specific NPs, or they are discourse-new yet hearer-old, which makes other candidates in the sentence more suitable topics, given Jacobs' three criteria quoted above in Section 2.1. See Bourgonje and Stede (to appear) for details.

5 Summary and Outlook

The PCC 2.1 release is now available both as raw corpus with the original annotation tool output files[10], and ready for inspection in the ANNIS3 interface[11]. Its multi-layer architecture makes it a suitable resource for corpus analyses of different types, easily combining information from different annotation layers through the ANNIS3 query engine; we gave three examples of queries in the AQL language. Another use case for annotated corpora is to serve as train/test data for specific applications, such as described in (Reitter, 2003) for RST parsing. Recently, Tuggener (2016) employed the PCC for testing his German coreference resolver. As a final example, in (Bourgonje and Stede, 2018), the connective layer has been exploited to train an automatic connective classifier. To establish the impact of parsing errors, the manually-annotated syntax trees (originating from a different annotation layer than the connective one) have been used.

While more annotation layers are not imminent right now, one step we foresee is to add information on sentence semantics and pragmatics, in order to build a further bridge between sentential syntax and the discourse-level annotations. This may concern 'semantic entity types' (as annotated by Becker et al. (2016) on a different German corpus) as well as expressions of different kinds of subjectivity.

One potential extension of an existing layer addresses the discourse connectives. As mentioned in Section 2, the connective layer now contains explicit relations only. To expand this to also cover implicit relations (and potentially also alternative lexicalizations, entity relations and 'no

[10] `http://angcl.ling.uni-potsdam.de/resources/potsdam-commentary-corpus-2.1.zip`
[11] `https://korpling.org/annis3/#_c=cGNjMi4x`

relations', as annotated in the PDTB), currently the RST layer can be exploited: For two adjacent text spans for which no explicit relation has been annotated, an RST relation may exist from which an implicit relation at the shallow level can (semi-)automatically be derived. However, RST and PDTB use different sets of relations, and – more importantly – it can be interesting to actually compare annotator's assignments of coherence relations (i) with and (ii) without the requirement of an overall spanning structural representation. Hence, we plan to work toward a complete, PDTB-style layer of annotation.

Finally, work is ongoing to expand the text base. A currently non-public part of the PCC contains editorials from *Der Tagesspiegel*, with some of the annotation layers mentioned above. Following an agreement with the publisher we hope to be able to release a bigger corpus (albeit not with all the layers) in the not too distant future.

References

Becker, M., Palmer, A., and Frank, A. (2016). Argumentative texts and clause types. In *Proceedings of the Third Workshop on Argumentation Mining*, Berlin. Association for Computational Linguistics.

Bourgonje, P. and Stede, M. (2018). Identifying explicit discourse connectives in German. In *Proceedings of the 19th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2018)*, pages 327–331, Melbourne, Australia. Association for Computational Linguistics.

Bourgonje, P. and Stede, M. (To appear). Topics and subjects in German newspaper editorials: A corpus study.

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER treebank. In *Proc. of the Workshop on Treebanks and Linguistic Theories*, Sozopol.

Cook, P. and Bildhauer, F. (2013). Identifying 'aboutness topics': two annotation experiments. *Dialogue and Discourse*, 4(2):118–141.

Danlos, L., Rysova, K., Rysova, M., and Stede, M. (2018). Primary and secondary discourse connectives: definitions and lexicons. *Dialogue and Discourse*, 9(1):50–78.

Dipper, S., Götze, M., Stede, M., and Wegst, T. (2004). Annis: A linguistic database for exploring information structure. In *Interdisciplinary Studies on Information Structure*, ISIS Working papers of the SFB 632 (1), pages 245–279.

Hoek, J., Evers-Vermeul, J., and Sanders, T. (2018). Segmenting discourse: Incorporating interpretation into segmentation? *Corpus Linguistics and Linguistic Theory*, 14(2):357–386.

Jacobs, J. (2001). The dimensions of Topic–Comment. *Linguistics*, 39(4):641–681.

Krasavina, O. and Chiarcos, C. (2007). PoCoS: The Potsdam Coreference Scheme. In *Proc. of the Linguistic Annotation Workshop (LAW) at ACL-07*, Prague.

Krause, T. and Zeldes, A. (2016). ANNIS3: A new architecture for generic corpus query and visualization. *Digital Scholarship in the Humanities*, 31.

Mann, W. and Thompson, S. (1988). Rhetorical Structure Theory: Towards a functional theory of text organization. *TEXT*, 8:243–281.

Matthiessen, C. and Thompson, S. (1988). The structure of discourse and 'subordination'. In Haiman, J. and Thompson, S., editors, *Clause combining in grammar and discourse*, pages 275–329. John Benjamins, Amsterdam.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., and Webber, B. (2008). The Penn Discourse Treebank 2.0. In *Proc. of the 6th International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco.

Reitter, D. (2003). Simple signals for complex rhetorics: On rhetorical analysis with rich-feature support vector models. *LDV Forum*, 18(1/2):38–52.

Sennrich, R., Schneider, G., Volk, M., and Warin, M. (2009). A new hybrid dependency parser for German. In Chiarcos, C., de Castilho, R. E., and Stede, M., editors, *From Text to Meaning: Processing Text Automatically. Proceedings of the Biennial GSCL Conference 2009*, pages 115–124, Tübingen. Narr.

Stede, M. (2004). The Potsdam Commentary Corpus. In *Proc. of the ACL Workshop on Discourse Annotation*, pages 96–102, Barcelona.

Stede, M., editor (2016). *Handbuch Textannotation: Potsdamer Kommentarkorpus 2.0*, volume 8 of *Potsdam Cognitive Science Series*. Universitätsverlag, Potsdam.

Stede, M. and Mamprin, S. (2016). Information structure in the Potsdam Commentary Corpus: Topics. In *Proc. of the Ninth International Conference on Language Resources and Evaluation (LREC)*, Portorož, Slovenia. European Language Resources Association (ELRA).

Stede, M. and Neumann, A. (2014). Potsdam Commentary Corpus 2.0: Annotation for discourse research. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 925–929, Reikjavik.

Tuggener, D. (2016). *Incremental Coreference Resolution for German*. PhD thesis, University of Zurich, Faculty of Arts.

Parsed Annotation with Semantic Calculation

Alastair Butler[1]*, Stephen Wright Horn*[2]

(1) Faculty of Humanities and Social Sciences, Hirosaki University
(2) Theory and Typology Division, National Institute for Japanese Language and Linguistics

ajb129@hirosaki-u.ac.jp, horn.s.w@ninjal.ac.jp

ABSTRACT

This paper describes a corpus building program implemented for Japanese (Contemporary and Old) and for English. First, constituent tree syntactic annotations defined to describe intuitions about sentence meaning are added to the texts. The annotations undergo tree transformations that normalise the analyses while preserving basic syntactic relations. The normalisation takes the parsed data for what are very different languages to a level where language particulars have a common interface to feed a semantic calculation. This calculation makes explicit connective, predicate, argument, and operator-binding information. Such derived information reflects universal principles of language: headedness, argumenthood, modification, co-reference, scope, etc. The semantic calculation also sets some minimal conditions for well-formedness: that predicative expressions are paired with subjects; that pro-forms have retrievable referents; that a constituent is associated with at least one grammatical function, etc. Annotators confirm and correct the source annotation with the aid of a visualisation tool that integrates the calculated output as overlaid dependency links. In this way annotators ensure that their interpretation of a text is correctly represented in its annotation. Furthermore, the integration of results from the semantic calculation makes it possible to establish multiple layers of grammatical dependencies with a minimum of invested annotation work.

KEYWORDS: Parsed corpus, Sentence and discourse meaning, Normalisation, Grammatical dependencies, Discourse referents, Visualisation.

1 Introduction

This paper presents the techniques behind a parsed annotation approach that has led to the creation of corpus resources with syntactic and semantic information for languages, including:

- Contemporary English (TSPC; 7,026 trees; 87,182 words; `http://www.compling.jp/ajb129/tspc.html`),
- Contemporary Japanese (NPCMJ; 20,425 trees; 315,200 words; `http://npcmj.ninjal.ac.jp/interfaces`), and
- Old Japanese (MYS97; 159 trees; 2,549 words; `http://www.compling.jp/mys97`).

These corpus resources offer instantiations of the goal to provide fully searchable representations of texts that are described according to interpretations of the meanings of sentences. This involves analysing texts into units and accounting for how the units are composed into meaningful expressions. Units are related by either grammatical functions or semantic relations. These two types of relations are together referred to as 'dependencies'. The annotation uses unit types, structural relations, and only as a last resort, indices, to describe how dependencies are established. A full description of a sentence tells the story of how information flows through structure to compose complex meanings. In order to achieve this, the annotation uses familiar grammatical categories to name units in an economical way, following organising principles.

The principles depend on two fundamental points: (1) an adequate description of the grammar, and (2) the ability to calculate a semantic representation. In essence, we take a text, divide it up into utterances (sentences), and decide on interpretations. Next we organise the parts of the sentence to show how the parts combine in principled ways to compose the meaning we arrive at in our interpretation. This takes the form of a tree with labeled nodes (a structure). The structural assignments in the annotation are associated with a finite set of defined relationships for interpreting meaning (dependencies), so that through transformations between representations that capture the structure, we can reach alternative instantiations of the structure that can be more explicit articulations of meaning components and their comprising dependencies.

Notably, the parse annotated tree structure will be transformed into expressions of a logical language, comprised of: discourse referents, operators, predicate relations, and logical connectives. A logical expression is reached by undertaking a recursive calculation that receives normalised annotation as input. We will refer to this as the 'semantic calculation'. If the logical expression derived from a given sentence annotation corresponds to the original interpretation, this is one kind of confirmation that the structure assigned to that sentence is correct, and serves as feedback to the annotation process.

This paper is organised as follows. In section 2 the basics of the annotation schema are described: word segmentation and part-of-speech tagging, labelled bracketing of segments and phrases, and the specification of grammatical function and clause linkage type. Then in section 3 we describe the first step from a parsed annotation to reach a semantic calculation. This step is normalisation, an information-preserving structural transformation in which language-specific syntactic constructions are rewritten into configurations built out of a reduced set of structures and categories. In section 4 we briefly sketch the remaining steps taken to reach a formal expression with the properties of a Discourse Representation Structure (Kamp and Reyle 1993). In section 5 we show how the information distilled in the logical expression analysis is integrated into the parsed annotation with indexing derived from the calculated discourse referents, becoming the basis for a graphic user interface for visualising sentence and discourse level dependencies. Section 6 relates the described method to alternative approaches for corpus development. Section 7 concludes the paper.

2 Basic annotation

This section presents an approach to syntactic analysis adopting parsing schemes influenced by the Penn Historical Corpora scheme (Santorini 2010) and the SUSANNE scheme (Sampson 1995). The Penn Historical Corpora scheme has informed the 'look' of the annotation, with tag labels familiar from mainstream linguistics, and the CorpusSearch format (Randall 2009). From the SUSANNE scheme, there is adoption of construction analysis, functional and grammatical information, and methods for representing complex expressions. The annotation also contains plenty that is innovative, including annotation to explicitly mark exceptional scope information (not described here for reasons of space) and information to resolve both inter and intra sentential anaphoric dependencies from a discourse perspective. Annotation is carried out with a general text editor (in practice, vi or emacs) on text files.

We can describe the overall annotation in terms of different layers of information:

1. Word segmentation and classification
2. Labelled bracketing of segments
3. Indicating grammatical function
4. Clause linkage types
5. Null elements
6. Scope marking
7. Tracking anaphora

This ordering of layers represents a loose hierarchy of informational detail, such that a 'higher' layer cannot be added unless a 'lower' (i.e. more basic) layer is already available: e.g., 2 is a pre-requisite for 3. The goal is a representation that is rich enough to allow the automatic calculation of argumenthood, modification, scope, and co-reference in discourse having little or no recourse to overt indexing. In practice, decomposing words into bindings and recasting grammatical dependencies as the management of discourse referents for those bindings can be accomplished with a surprisingly modest inventory of grammatical categories and structural configurations annotated on the source data, provided that all relations in a set of parsed trees are defined under a sufficiently developed system of calculation. The remainder of this section introduces the basics for establishing such annotation.

2.1 Word segmentation and classification

A continuous segment is a word and each word is assigned a part-of-speech (POS) tag—in practice, the label of an open bracket '(LABEL'. A POS tag can be complex, having at least a core label, but potentially including extensions separated by hyphens, and in certain cases, semantic information in curly brackets set off with a semicolon. In (1), being especially relevant for parsing Japanese, particles (P) receive extensions to mark distinct contributions:

(1)
```
P-COMP  - complementiser
P-CONN  - connective with coordination or subordination
P-FINAL - sentence final, e.g., marker of clause type
P-OPTR  - operator
P-ROLE  - marker of grammatical (syntactic or semantic) role
```

In (2), a determiner is coded for definiteness by adding semantic information:

```
(2)  (D;{DEF} sono)    "that"
```

The POS tags for a language may be sufficiently fine-grained to mark syntactic features. For example, the distinction between a common noun and a 'formal noun' used in root contexts to indicate aspect in Japanese is specified as in (3):

```
(3)  (N tokoro)        "place"
     (FN tokoro)       "occasion, instant"
```

Note that for annotating Japanese, grammatical number is not indicated by a specialised POS tag, in contrast to English seen in (4):

```
(4)  (N spoon)         singular
     (NS spoons)       plural
```

2.1.1 Multi-word expressions

A sequence of words which behaves syntactically in an idiomatic way (e.g., *ni muke te* 'with a view towards', *ga hayai ka* 'no sooner than', *sore ni shi te mo* 'irregardless') is 'chunked' into a single segment and tagged by using the word tag appropriate for the sequence as a whole, as illustrated in (5).

```
(5)  (P-ROLE nimukete)
     (P-CONN gahayaika)
     (CONJ sorenishitemo)
```

That is, such word sequences are treated as single units for the purposes of parsing.

2.2 Labelled phrase bracketing

Bracketing is a way of grouping together segments which are recognised as having a syntactic integrity as units of various kinds (sentences, clauses, phrases, words). Typically, when a $word_1$ combines with a $phrase_1$ to form a more complex $phrase_2$, $word_1$ is said to be the head of $phrase_2$. Let's consider providing syntactic analysis for (6).

(6) daijoobu ？sensei ！to odoroi ta koe de bokura wa kii ta 。
OK ? teacher ! COMP surprise PAST voice INSTR we TOP ask PAST .
' "Teacher! Are (you) OK?" we asked (him) in surprised voices.'

In example (6), represented with bracketing in (7) below, pronoun *bokura* 'we' is a segment that, being unmodified, forms a phrase by itself. As a phrase, it can combine with a head, in this case topic particle *wa*. Particle *wa* projects a topic particle phrase *bokura wa*, which relates as a whole to the verb *kii* 'ask'. In contrast, *sensei* 'teacher', is a vocative phrase, relating to an immediately preceeding sentence, *daijoobu* ？ 'Are you OK?' by virtue of co-reference. These two units combine to form an utterance marked by a complementiser *to*. The resulting unit appears at the same structural level as the particle phrase *bokura wa*, where it too relates to the verb *kii*. The outermost brackets indicate that the whole complex utterance is a unit. Thus the verb *kii* 'ask' is part only of the whole utterance in example (7), and as such is the head of the largest sentence unit.

```
(7)  ( ( ( ( daijoobu) ？
           ( sensei) ！) to)
       ( ( ( odoroi ta)
           koe)
         de)
       ( bokura wa)
       kii ta
       。)
```

This layer of bracketing indicates the composition of non-terminal syntactic units (or constituents) such as Noun Phrase, labelled NP, Preposition/Postposition Phrase, labelled PP, Clause, labelled IP, and Complementiser Phrase, labelled CP, etc. For example, adding phrase-level labels to (7) above yields the labelled analysis of (8):

```
(8)  (IP (CP (CP (IP daijoobu) ？
                 (NP sensei) ！) to)
           (PP (NP (IP odoroi ta)
                   koe)
               de)
           (PP (NP bokura) wa)
           kii ta
           。)
```

For phrase-level annotation in Japanese, the following basic categories are assumed:

- Clause unit (which can be the utterance-level) (IP, CP, FRAG)
- Noun phrase (NP)
- Postposition phrase (PP)
- Adverb phrase (ADVP)

Annotation for English adds one further phrase type:

- Adjective phrase (ADJP)

Including the POS labels for terminal nodes completes the basic structure of a parsed tree:

```
(9)  (IP (CP (CP (IP (ADJN daijoobu)) (PU ？)
                 (NP (N sensei)) (PU ！))
             (P-COMP to))
           (PP (NP (IP (VB odoroi)
                       (AX ta))
                   (N koe))
               (P-ROLE de))
           (PP (NP (PRO bokura)) (P-OPTR wa))
           (VB kii)
           (AXD ta)
           (PU 。))
```

2.3 Indicating grammatical function

The grammatical function which relates a unit to the element with which it combines is identified by either:

1. the combination of the head plus the structural position of that unit, (e.g., a P-ROLE heading a PP may specify the function of that PP with respect to a predicate),
2. the node description of that unit having hyphenated extensions with information about grammatical function, such as -SBJ for subjects, -LOC for locational adjuncts, etc., or
3. having 1 supplemented by 2.

Example (10) illustrates scenario 1 with a PP headed by instrumental case marker `(P-ROLE de)` at the matrix clause level (directly below IP-MAT). The information supplied by the terminal node `de` together with the part of speech tag P-ROLE is considered sufficient for determining the grammatical function of the constituent.

Example (10) illustrates scenario 2 with its PP-SBJ at the clause level. As an operator particle, the 'toritate' particle *wa* can mark topichood, contrast, or focus of negation, for example, but not grammatical role, so the function of the PP with respect to the head of the whole sentence is indicated by the combination of (i) the sibling relationship between the PP and the head of the whole sentence, and (ii) the extension -SBJ.

(10)
```
(IP-MAT (CP-THT (CP-QUE (IP-SUB (ADJN daijoobu))
                        (PU ？)
                        (NP-VOC (N sensei))
                        (PU ！))
                (P-COMP to))
        (PP (NP (IP-EMB (VB odoroi)
                        (AX ta))
                (N koe))
            (P-ROLE de))
        (PP-SBJ (NP (PRO bokura))
                (P-OPTR wa))
        (VB kii)
        (AXD ta)
        (PU 。))
```

At the clause level, all units locally relating to (i.e., sibling to) the head must have grammatical function information. Grammatical function information can be either syntactic (e.g., -SBJ for subject) or semantic (e.g., -LOC for location). In contrast, as a complement within a PP, an NP need not have an extension. For utterances that consist of a question, the tag CP-QUE is used. In (10) this unit combines with the `(P-COMP to)` to form a complementiser phrase CP-THT. This in turn combines with `(VB kii)`. Note that the heads of sentences (i.e., predicates) can be analytic. Here the past tense morpheme `(AXD ta)` forms one part of a complex verbal syntagm.

At this point we can begin tracking how the syntactic information is ultimately interpreted by comparing the interpretation encoded in the annotation (here, specifically, the intuition that `bokura wa` 'we' functions as the subject of the predicate `kii_ta` 'ask') with the formula output of the semantic calculation (11):

(11)
```
1  ∃ TEACHER[8] STUDENTS[5] EVENT[7] EVENT[9] de[6] THT[1].(STUDENTS[5] = bokura
2    ∧ to(THT[1],QUEST ∃ TEACHER[3] EVENT[4] TEACHER[2].(sensei(TEACHER[2])
3            ∧ TEACHER[3] = TEACHER[2]
4            ∧ daijoobu(EVENT[4],TEACHER[3])))
5    ∧ koe(de[6],odoroi_ta(EVENT[7],STUDENTS[5]))
6    ∧ TEACHER[8] = *pro*
7    ∧ past(EVENT[9])
8    ∧ kii_ta(EVENT[9],STUDENTS[5],_,TEACHER[8],THT[1]) ∧ de(EVENT[9]) = de[6])
```

In (11) we see a sorted discourse referent `STUDENTS[5]` (that is, referent `[5]` of sort `STUDENTS`) under existential quantification (discourse closure) and equated with pronoun `bokura` in line

1. In addition STUDENTS[5] is one of the arguments in a binding with predicate kii_ta in line 8. In addition, STUDENTS[5] appears as the argument of odoroi_ta in line 5.

Note that the basic annotation in (10) is not yet a complete mark-up of (6) with regard to the data model for the approach. More layers of information have to be added before a formula such as (11) can be generated. A complete annotation for (6) can be seen in (12).

(12)
```
(IP-MAT (CP-THT (CP-QUE (IP-SUB (NP-SBJ;{TEACHER} *pro*)
                                (ADJN daijoobu))
                        (PU ？)
                        (NP-VOC;{TEACHER} (N sensei))
                        (PU ！))
                (P-COMP to))
        (PP (NP (IP-EMB (VB odoroi)
                        (AX ta))
                (N koe))
            (P-ROLE de))
        (NP-OB2;{TEACHER} *pro*)
        (PP-SBJ;{STUDENTS} (NP (PRO bokura))
                           (P-OPTR wa))
        (VB kii)
        (AXD ta)
        (PU 。 ))
```

The need to feed a semantic calculation places requirements on the annotation. Trees are connected graphs with the first labelled branching node taking its label from a set of utterance types (Exclamation CP-EXL, Imperative IP-IMP, Fragment FRAG, etc.). In general an IP is defined as a nexus of a predicate (e.g., a verb, adjective, or copular expression) and at minimum a subject. The meaning of the predicate conditions the realization of arguments and modifiers. Elements that are not expressed overtly in the text may be assigned nodes with null elements, or they may be left empty to inherit the contribution of a non-local argument position. Elements combining with predicates must be specified for grammatical or semantic role. Pronominal elements (such as *bokura* in (10)) must have their reference resolved with 'sort' information. For example, an annotator will add '`;{STUDENTS}`' to the overt pronoun in (12). Floating quantifiers must be associated with target constituents. Relative clauses must contain a trace.

In general, constructions are assigned structures using a limited inventory of annotation conventions (labels, and indices). The structures are defined so that language-specific details can be differentiated, while at the same time basic dependencies can be extracted. In practice, an elaborated system of defaults is needed in order to cover the data in natural language to some degree. The data model employed here sets enough requirements to allow the semantic calculation to arrive at unambiguous results for argumenthood, modification, and anaphoric relations. Achieving this density of dependencies allows the generation of an output that can be examined and corrected for accuracy.

3 Normalisation

So far we have seen how applying the annotation produces a language-specific syntactic analysis of the source text. We now turn to the question of how to feed the information encoded in the annotated tree into the semantic calculation mechanism (sketched in section 4) to output a form in which grammatical relations are expressed as logical relations. This first step consists of rewriting language-specific structures so that information is preserved about the dependencies and the lexical material while taking a generalised form that can apply to any language. This

is called 'normalising' the annotation. We illustrate the process with (6), using its completely annotated form (12) from the previous section as the starting input.

Normalisation reduces the number of node labels, replaces items denoting basic grammatical roles with offset elements headed by labels specifying those roles, migrates some label information into terminal nodes as predicates denoting grammatical categories, and concatenates any terminal nodes which have had the information from their part of speech information discharged in generated predicates. The normalisation of (12) is shown in (13).

(13)
```
( (IP-MAT (ACT past)
          (CP-THT (C to)
                  (CP-QUE (PU ？)
                          (PP (P-ROLE VOC)
                              (NP (SORT *TEACHER*)
                                  (N sensei)))
                          (PU ！)
                          (IP-SUB (PP (P-ROLE ARG0)
                                      (NP (SORT *TEACHER*)
                                          (PRO *pro*)))
                                  (VB daijoobu))))
          (PP (P-ROLE ARG0)
              (P-OPTR wa)
              (NP (SORT *STUDENTS*)
                  (PRO bokura)))
          (PP (SORT de)
              (P-ROLE de)
              (CP-ADV (C koe)
                      (IP-SMC (VB odoroi_ta))))
          (PP (P-ROLE ARG2)
              (NP (SORT *TEACHER*)
                  (PRO *pro*)))
          (VB kii_ta)
          (PU 。))
  (ID 6_misc_discourse_3;JP))
```

By comparing (13) with (12), it can be seen that the NP-SBJ tag in the source annotation is changed into an NP tag which is placed under a PP projection headed by a `(P-ROLE ARG0)` element. All arguments are converted to PP projections with role information appearing as a terminal string (effectively to function as the information for how the argument's discourse referent will be linked to the predicate the argument serves to bind in a resulting logical expression). For underdetermined elements such as `(PP (NP (IP-EMB (VB odoroi) (AX ta)) (N koe) (P-ROLE de))`, the information carried by the particle is copied into the terminal node position of a `SORT` element. Part-of-speech tags are regularised in other ways as well. For example, `daijoobu` is a 'na-adjective' (a category peculiar to Japanese), but its ADJN tag is regularised to VB, which is the category for all clause level predicates after normalisation. Some information about basic grammatical categories is migrated from node labels into offsets. For example, the part of speech label for the past tense morpheme AXD triggers the creation of an ACT tag with a predicate `past` to retain the tense information. (Note that the past tense morpheme, stripped of its part of speech tag, is concatenated to the free morpheme immediately preceding it.) For pronouns, the 'sort' information that resolves their reference is made into an off-set element as well.

Normalisation renders all the information relevant to the domain of discourse in forms that are defined under an intermediate formal language. The next step is to pass the normalised tree to a script that decomposes the lexical items into bindings, and recasts structural relations

as instructions about how to manipulate them. The intermediate language, then, is basically a set of interrelated rules that use the normalised information to generate predicates, binding names, and the sequences of discourse referents that binding names are assigned, plus various operations over and relations between these elements.

4 The semantic calculation

At the start of the calculation, there is a collection of 'discourse referents' which are read from the normalised input based on the occurrences of SORT nodes. Discourse referents are ultimately destined to be bound by operations of closure (e.g., ∃) in the overall resulting logical expression. The operations of closure are themselves either reflexes of quantification instructions from the normalised input, or arise because there is discourse closure. During the runtime of the calculation, the collected discourse referents are released as content for argument slots of predicates that populate the resulting logical expression. The exact makeup of argument slots (valencies) for a given predicate is typically left unspecified by the input. At the point in the calculation when the predicate is reached, the availability of accessible discourse referents is established. The predicate's sensitivity to what is accessible determines the arguments for the predicate. More information about the semantic calculation is available at: `http://www.compling.jp/ajb129/ts.html`. As a result of calculation, (13) is turned into the formula analysis seen as (11) of section 2.3.

5 Treebank Semantics Visualisation Tool

With a formula rendering, such as (11) of section 2.3, relationships are expressed by sorted discourse referents appearing in multiple contexts, or relating to other discourse referents (e.g., as being identical to other discourse referents). Such relationships can be re-expressed through indices shared between nodes in a tree structure. Such indices, being derived from the source annotation in the first place, are in principle a redundant notational variant rather than an addition of information. Nevertheless, they are convenient as loci for adding information about dependencies between constituents in tree structures. The dependencies derived from the annotation of (6), rewritten as indices, can be embedded back into the source phrase structure tree annotation to yield (14):

(14)

```
(IP-MAT (CP-THT;<THT[1]> (CP-QUE (IP-SUB (NP-SBJ;<TEACHER[3]>
                                             (PRO pro;{,TEACHER[2],}))
                                         (ADJN;<,TEACHER[3]@ARG0,EVENT[4]@EVENT,>
                                             daijoobu))
                                 (PU ？)
                                 (NP-VOC;<TEACHER[2]>
                                         (N;<,TEACHER[2]@h,> sensei))
                                 (PU ！))
                           (P-COMP;<,THT[1]@FACT,> to))
        (NP-OB2;<TEACHER[5]> (PRO pro;{,}))
        (PP-SBJ;<ENTITY[6]> (NP (PRO bokura;{,})) (P-OPTR wa))
        (VB;<,ENTITY[6]@ARG0,TEACHER[5]@ARG2,THT[1]@THT,EVENT[7]@EVENT,> kii)
        (AXD ta)
        (PU 。))
```

This 'indexed' view gives a view of the tree structure with indexing information that specifies

argument relationships and antecedence relationships. Argument dependencies are marked on the label of the target predicate as sets of index/grammatical function pairs. Antecedence relationships (including big PRO, as the terminal `*PRO*;{GROUP[1]}`) are spelled out using discourse referents typed according to 'sort' information.

This 'indexed' view explicitly encodes grammatical dependencies that the original annotation had left implicit. The indexing makes the following kinds of contributions:

- Indexing given the form '`<discourse_ref>`' marks a node that serves as an argument for a predicate.
- The arguments that a predicate takes are marked on the pre-terminal node for the predicate with a '`<,...,discourse_ref@role,..,>`' format, with '`discourse_ref`' providing information to locate the argument and '`role`' stating the argument role.
- Control and trace information is presented with the format '`{discourse_ref}`', that is, specifying a single obligatory antecedent.
- Pronominal information is presented with the format '`{,discourse_ref,...,}`', that is, specifying potentially multiple antecedents.

In this 'indexed' view the dependencies that obtain between constituents within a given tree are fairly easy to read, but for checking discourses over multiple trees, display in a graphic user interface is preferable. To this end, a web browser based tool was developed with a tree-drawing program and numbered nodes called the Treebank Semantics Visualisation Tool (TSVT). The components that make up the tool are available at: `http://www.compling.jp/ajb129/view.html`. The derived dependencies are displayed in various ways depending on their type. Figure 1 below presents the GUI image for the indexed view of (6) in the original Japanese script.

Local dependencies are indicated by integer/role pairs underneath the target predicate. For example, in figure 1 below, the predicate node `(VB 聞い)` —romanised *kii* 'ask'— has below it a set of pairs specifying constituents and their roles respective to *kii* 'ask':

- `243-arg2` (node no. 243, indirect object)
- `230-で` (node no. 230, instrumental particle 'de')
- `239-arg0` (node no. 239, subject)
- `217-tht` (node no. 217, clausal complement)

Note that ATB extraction relationships are drawn with the same notational convention as local relationships.

Control relationships are indicated by dotted lines connecting the node number of an antecedent constituent with a shared secondary node number for a derived subject position. For example, in figure 1 below the subject argument node `(PP-SBJ (NP (PRO 僕ら)) (P-OPTR は))` —romanised *bokura wa* 'we'— is in a control relationship with a derived subject position in a noun-modifying clause IP-EMB. The derived subject position is node no. 233, but carries a secondary node number matching that of its antecedent: no. 239, and connecting with it via a dotted line.

Anaphoric relationships are drawn with the same notational convention as control relationships. For example, in figure 1 below, the null subject in IP-SUB is co-referential with the post-posed vocative phrase `(NP-VOC (N 先生))` —romanised *sensei* 'teacher'. The node number of the antecedent (no. 225) is connected to a matching secondary number marked on the pronominal

constituent. In this respect the role of context in the interpretation of meaning becomes apparent. For example, (PP-SBJ (NP (PRO 僕ら)) (P-OPTR は)) (no. 239) and (NP-OB2 (PRO pro)) (no. 243) carry secondary numbers from a discourse referent fragment introduced at the beginning of a longer discourse (no.s 7 and 3, respectively). Additionally, an exophoric antecedent preceding (6) supplies the reference for a pronoun in a tree following (6), showing up as a dotted line passing uninterrupted through the tree in figure 1.

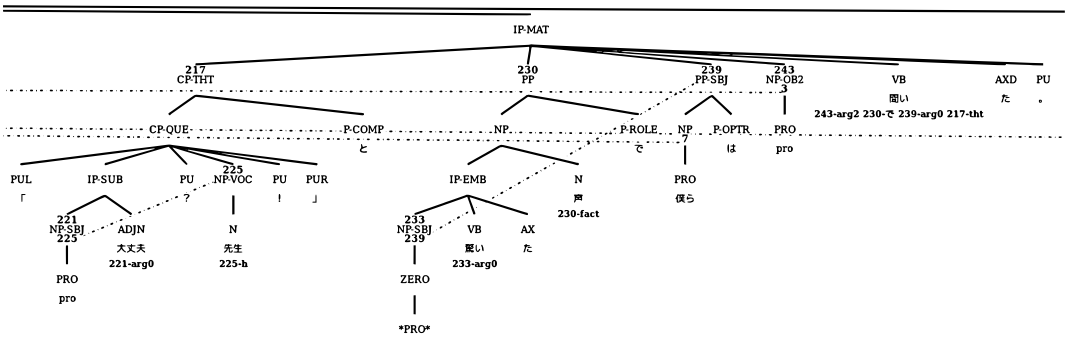


Figure 1: TSVT visualisation of (6)

The TSVT is able to provide an output at the point where the density of dependencies encoded in the source parse supply the minimum necessary information to account for the requirements/contributions of lexical items appearing in the source parse. Once this critical mass of information is reached, the tool can now assist in the establishment of additional dependencies that, while not stipulated by the grammar, are nonetheless crucial to meaning interpretation. Annotators can modify bracketed trees and see the results of the semantic calculation instantly using the TSVT. Corrections that have impact on the semantic calculation typically include changing attachment sites for constituents, changing the specification for clause linkage, the addition or subtraction of null pronouns, and the addition of 'sort' information to noun phrases. Overt indices are rarely used in the annotation, the structural assignments given are usually common patterns, and exceptions are typically listed in the annotation documentation. In short, the predictable behaviour of the semantic calculation allows annotators to use a finite set of strategies to represent all the basic dependencies involved in the interpretation of sentence meaning in discourse context.

6 Comparison with other approaches

The techniques described in this paper are complementary to the classic treebank methods of building parse trees as developed at Lancaster (e.g., Garside, Leech and Sampson 1987, Sampson 1995) and Penn (e.g., Marcus, Santorini and Marcinkiewicz 1993, Kroch, Santorini and Diertani 2010). To this, the approach adds a semantic calculation dimension. In this regard, Butler and Yoshimoto (2012) and Kiselyov (2018) are closely related approaches.

Alternative wide-coverage systems with semantic components have been developed by other means. One prominent branch has involved boot-strapping from tree based corpora to produce resources within categorial grammar frameworks, notably, CCGbank (Hockenmaier and Steedman 2005) and TLGbank (Moot 2015). The CCGbank approach has been extensively developed with the Groningen Meaning Bank (Basile et al. 2012) and its multi-language variant,

the Parallel Meaning Bank (Abzianidze et al. 2017), where there is improvement of source CCG annotation to allow for an interleaving of grammar development and corpus annotation in a beneficial cycle. In this regard the major difference with the current approach is the base input ('Bits of Wisdom' versus constituent trees), as both approaches create Discourse Representation Theory analysis (Kamp and Reyle 1993).

With corpus annotation relying on a beneficial cycle of analysis results, there is a clear link with approaches to corpus production that have emerged out of full grammar development programs, e.g., DeepBank (Flickinger et al. 2012) based on HPSG, and NorGramBank (Dyvik et al. 2016) based on LFG. Such programs have led to the creation of high coverage corpora, and have included semantic levels of analysis, and have been applied cross-linguistically. In comparison, the current approach is significantly 'lighter' in the sense of not requiring the development of a full grammatical system for the annotation to make progress.

7 Conclusion

Linking structure and semantics sets constraints on how the whole corpus is designed. Starting from the semantic requirements, at a very basic level, discourse referents need to be introduced as parts of resulting logical expressions, and there must also be the means for introduced discourse referents to find their way to the appropriate argument slots of predicates, where predicates are used to instantiate the contributions of nouns, verbs, adjectives, adverbs, etc. Such introductions and their subsequent management have to be linked to the structures in the corpus. Needing to establish such links can actually simplify the structure of the corpus. This is a reflection of the fact that languages in general have fixed ways of keeping track of language components. In grammar, we see these facts as the reach (or lack of reach) of an argument dependency through structure, the marking of definiteness and specificity to project scope, accessibility conditions on anaphoric reference, etc.

The semantic calculation, together with the application that allows its results to be displayed (the TSVT), make it possible to establish multiple layers of grammatical dependencies with a minimum of annotation. The ability to re-integrate derived relationships into parsed trees makes it possible, for example, to do statistical analyses on pronominalisation patterns and co-valuation (for example, topic persistence, donkey anaphora, some types of periphrasis, etc.). When paired with an exhaustive analysis of grammatical relations in context, adding an additional layer of semantic roles is informative not only for lexical semantics, but also for studies on the interaction of semantic and grammatical roles (e.g., case frame theory). With the semantic calculation, fleshed out lexical profiles for phrasal heads can be generated. These can be used, for example, to identify null positions, which in turn can add to the possibilities for annotating cohesive relations in texts. With the systematic enrichment of annotation capturing the intuitions of native speakers, a data-driven description of language at a high level of abstraction becomes a real possibility.

Acknowledgements

We are grateful to three anonymous reviewers who gave extremely valuable feedback to improve the paper. Support was received from JSPS Grant-in-Aid for Scientific Research KAKENHI grants 15K02469 and 18K00560. Support also came from the "Development of and Research with a Parsed Corpus of Japanese" collaborative project based at the National Institute for Japanese Language and Linguistics. We thank all project members for their help and advice.

References

Abzianidze, Lasha, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The parallel meaning bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages Valencia, Spain. 242–247.

Basile, V., J. Bos, K. Evang, and N.J. Venhuizen. 2012. Developing a large semantically annotated corpus. In *Proceedings of the 8th Int. Conf. on Language Resources and Evaluation*. Istanbul, Turkey.

Butler, Alastair and Kei Yoshimoto. 2012. Banking meaning representations from treebanks. *Linguistic Issues in Language Technology - LiLT* 7(1):1–22.

Dyvik, Helge, Paul Meurer, Victoria Rosén, Koenraad De Smedt, Petter Haugereid, Gyri Smørdal Losnegaard, Gunn Inger Lyse, and Martha Thunes. 2016. NorGramBank: A 'Deep' Treebank for Norwegian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3555–3562. Paris, France: European Language Resources Association (ELRA).

Flickinger, Dan, Valia Kordoni, and Yi Zhang. 2012. DeepBank: A dynamically annotated treebank of the Wall Street Journal. In *Proceedings of TLT-11*. Lisbon, Portugal.

Garside, Roger, Geoffrey Leech, and Geoffrey Sampson, eds. 1987. *The Computational Analysis of English: a corpus-based approach*. London: Longman.

Hockenmaier, Julia and Mark Steedman. 2005. CCGbank: User's manual. Tech. Rep. MS-CIS-05-09, Department of Computer and Information Science, University of Pennsylvania, Philadelphia.

Kamp, Hans and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer.

Kiselyov, Oleg. 2018. Transformational Semantics on a Tree Bank. In S. Arai, K. Kojima, K. Mineshima, D. Bekki, K. Satoh, and Y. Ohta, eds., *JSAI-isAI 2017*, vol. 10838 of *Lecture Notes in Computer Science*, pages 241–252. Heidelberg: Springer.

Kroch, Anthony, Beatrice Santorini, and Ariel Diertani. 2010. *The Penn-Helsinki Parsed Corpus of Modern British English (PPCMBE)*. Department of Linguistics, University of Pennsylvania. CD-ROM, second edition, (http://www.ling.upenn.edu/hist-corpora).

Marcus, Michell, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2):313–330.

Moot, Richard. 2015. A Type-Logical Treebank for French. *Journal of Language Modelling* 3(1):229–265.

Randall, Beth. 2009. CorpusSearch 2 Users Guide. (http://corpussearch.sourceforge.net/CS-manual/Contents.html).

Sampson, Geoffrey R. 1995. *English for the Computer: The SUSANNE Corpus and Analytic Scheme*. Oxford: Clarendon Press (Oxford University Press).

Santorini, Beatrice. 2010. Annotation manual for the Penn Historical Corpora and the PCEEC (Release 2). Tech. rep., Department of Computer and Information Science, University of Pennsylvania, Philadelphia. (http://www.ling.upenn.edu/histcorpora/annotation).

Data Conversion and Consistency of Monolingual Corpora: Russian UD Treebanks

Kira Droganova[1], *Olga Lyashevskaya*[2,3], *Daniel Zeman*[1]

(1) Charles University, Faculty of Mathematics and Physics, Prague
(2) National Research University Higher School of Economics, Moscow
(3) Vinogradov Institute of the Russian Language RAS, Moscow

{droganova,zeman}@ufal.mff.cuni.cz, olesar@yandex.ru

Abstract

In this paper we focus on syntactic annotation consistency within Universal Dependencies (UD) treebanks for Russian: UD_Russian-SynTagRus, UD_Russian-GSD, UD_Russian-Taiga, and UD_Russian-PUD. We describe the four treebanks, their distinctive features and development. In order to test and improve consistency within the treebanks, we reconsidered the experiments by Martínez Alonso and Zeman; our parsing experiments were conducted using a state-of-the-art parser that took part in the CoNLL 2017 Shared Task. We analyze error classes in functional and content relations and discuss a method to separate the errors induced by annotation inconsistency and those caused by syntactic complexity and other factors.

Keywords: Annotation consistency, Universal dependencies, Russian treebanks, Dependency parsing.

1 Introduction

The co-existence of several treebanks for one language, made by different teams and converted from different sources, within the Universal Dependencies (UD) platform (Nivre et al., 2016) is a key factor that makes data grow and attracts new contributors. A user can choose a treebank more appropriate for their needs, or combine data into one training set. However, the heterogeneous nature of treebanks in terms of their annotation policies, genres, etc. means that combining data is not straightforward.

During the last two decades surprisingly little attention has been paid to syntactic annotation consistency. Brants and Skut (1998) proposed using a statistical parser for the detection and labeling of phrase structures and specifying the functions of the words during annotation. The method is more of an annotation strategy rather than a mechanism for checking consistency within an already annotated treebank. Dickinson and Meurers (2003) described an approach of using n-grams for identifying errors in phrase structure trees in context: n-grams with annotations that vary in the same context signal inconsistency. Longer contexts are preferable. Kaljurand (2004) presented a tool for checking consistency by restructuring a treebank in a way that the focus is given to groups which can be formed by either words, POS-tags or syntactic labels. Statistics are extracted from the treebank for each group. After that, consistency is estimated using the context: the annotation of the group varies less in a wider context. The tool is restricted to the NEGRA format (Brants, 1997). A method independent of dependency representation was proposed by Boyd et al. (2008), who reused the concept of variation nuclei (Dickinson and Meurers, 2003) and extended it with context restrictions. Kulick et al. (2013) combined the concept of variation nuclei with the idea of decomposing full syntactic trees into smaller units, and applied it to the task of the evaluation of inter-annotator agreement. Another approach to consistency checking is the use of heuristic patterns. De Smedt et al. (2015) proposed this strategy to check the consistency of multi-word expressions within the Universal Dependencies (UD) treebanks with INESS-Search. Martínez Alonso and Zeman (2016) conducted a series of parsing experiments to determine the consistency of the AnCora Spanish and Catalan treebanks after their conversion to UD. One of the most recent works (de Marneffe et al., 2017) adapts the method proposed by Boyd et al. (2008) to the UD treebanks, tests the approach on English, French and Finnish UD treebanks and proposes an annotation tool to organize the manual effort required by the method. Another recent work by Alzetta et al. (2018) describes a method aimed at detecting erroneously annotated arcs in gold standard dependency treebanks by measuring the reliability of automatically produced dependency relations.

For the purpose of the analysis of syntactic annotation consistency within the four Russian treebanks in UD, we have chosen the method by Martínez Alonso and Zeman (2016). They conducted a series of monolingual, cross-lingual and cross-domain parsing experiments using attachment accuracies as the means to estimate consistency with other UD treebanks and between two treebanks of the same language (Spanish AnCora and 'Web' treebank, the latter now being officially labeled GSD). We consider the method useful not only for treebank contributors but also for external users, who need a simple measure to make a decision about which corpora can be jointly used in their experiments. Labeled attachment score (LAS) is a good candidate to serve as a simple criterion to find an optimal corpus mix for parsing experiments. On top of that, the method is suitable for languages with rich morphology and is context-independent; the latter is very important because the four Russian corpora comprise texts of different genres. The method proposed by de Marneffe et al. (2017)

is mostly designed for UD contributors and is not fully suitable for morphologically-rich languages[1]. However, we plan to conduct such experiment in the future.

2 UD Russian Treebanks

The UD release 2.2 includes four Russian treebanks: **SynTagRus, GSD** (Google Stanford Dependencies), **PUD** (Parallel Universal Dependencies) and **Taiga.** Only Taiga was annotated directly in the UD style; the other three treebanks were manually annotated under a different scheme and then automatically converted to UD. The original scheme of GSD and PUD (both annotated at Google) is arguably more similar to UD than SynTagRus.

2.1 UD_Russian-SynTagRus

The Russian dependency treebank, SynTagRus, was taken as the source. Until the other UD treebanks emerged, SynTagRus was the only human-corrected corpus of Russian supplied with comprehensive morphological annotation and syntactic annotation in the form of a complete dependency tree provided for every sentence (Boguslavsky et al., 2009; Dyachenko et al., 2015).

The treebank is built upon Melčuk's Meaning-Text Theory (Mel'čuk, 1981) and specifies a set of 67 dependency relations, 11 coarse-grained part-of-speech tags and a set of morphological features.

Currently the treebank contains over 1,000,000 tokens (over 66,000 sentences) belonging to texts from a variety of genres (contemporary fiction, popular science, texts of online news, newspaper and journal articles, dated between 1960 and 2016).

2.1.1 Key features of corpus development

We developed the conversion procedure using the original corpus statistics and the corpus description[2]. Clearly, the original SynTagRus annotation principles and the UD guidelines have important differences that should be explained in detail.

1. Following the UD principle that dependency relations hold primarily between content words, the tree structure has been transformed. The following nodes have been moved to dependent positions:
   (a) Prepositions (heads of prepositional phrases in SynTagRus);
   (b) Copulas and auxiliary verbs;
   (c) Coordinating conjunctions (coordinate clauses are connected via conjunctions in SynTagRus);
   (d) Subordinating conjunctions (subordinate clauses are attached via subordinating conjunctions in SynTagRus).
2. SynTagRus currently provides a set of 67 dependency relations. However, only 16 relations can be directly mapped to appropriate relations within the set of 37 UD dependency relations. The remaining majority of relations require additional information

[1]Their lemma-based approach fails on morphologically-rich languages. The results for the wordform-based approach seem promising even for morphologically-rich languages.

[2]`http://www.ruscorpora.ru/instruction-syntax.html`

concerning morphological information of the node, morphosyntactic information of the head and the dependents of that node due to differences between the original annotation and the UD guidelines:

(a) The original relations do not distinguish between core arguments and oblique dependents. The rule of thumb in Russian UD is that core arguments are bare noun phrases (without a preposition) in nominative, accusative, dative or genitive (although the latter two cases are slightly controversial and may be reconsidered in the future);

(b) The original relations do not encode information about clauses;

(c) Some of the relations in UD depend on part-of-speech tags, which is not always the case for original SynTagRus dependency relations.

3. Elliptic constructions are represented in SynTagRus by reconstructed tokens, which contain all the information a normal token would, except the word form. Regarding dependency structure, these tokens behave like normal nodes. This representation allows the development of conversion rules to generate both the basic and the enhanced UD layer.

   In the basic representation, reconstructed nodes are omitted and dependency paths that traverse them are contracted. Then the trees are rearranged according to the obliqueness hierarchy specified in the UD guidelines (Figure 1). In the enhanced representation, the reconstructed nodes are preserved.

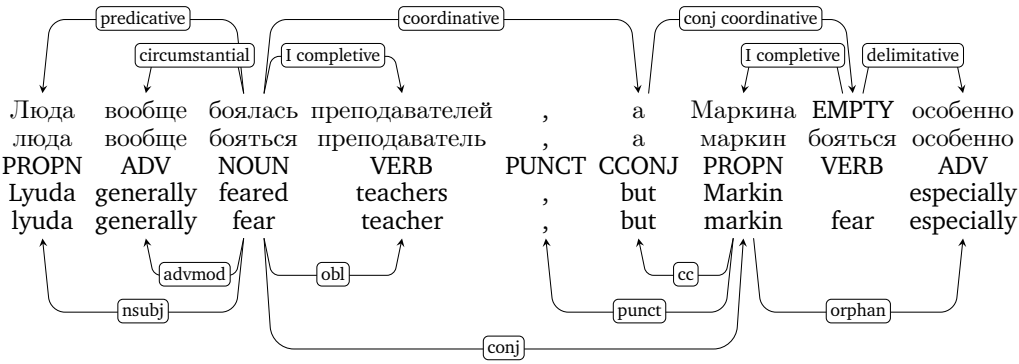


Figure 1: An example of a Russian elliptic sentence. The upper part shows the original SynTagRus dependency tree annotated with original dependency relations. The bottom part shows the UD style converted tree. Translation: “Lyuda generally feared teachers, but Markin especially.”

4. Multiword expressions are mainly re-analyzed using a manually created MWE dictionary, which includes POS tags, features and dependency relations. In SynTagRus, a multiword expression is usually placed into one node where words are separated by spaces. Therefore, it has just one part-of-speech tag, one set of features and one dependency label. Additionally, we use a rule that makes sure that the first node of the MWE is technically the head and all other member words are attached to it.

5. The UD guidelines specify a set of 21 morphological features, which can be extended by language-specific features. However, only 13 features were involved in the conversion process. The rest either do not apply to Russian or could not be reliably derived from the features in SynTagRus.

6. For most of the 11 part-of-speech tags in SynTagRus the conversion mapping is pretty straightforward. However, several heuristics were developed for tricky cases:

   (a) SynTagRus does not distinguish between content verbs and auxiliary verbs. The AUX tag is always assigned to the lemma "быть" (to be);

   (b) Determiners are converted using closed-class lists since they are annotated as adjectives in SynTagRus;

   (c) Proper nouns do not have a special tag in SynTagRus. Therefore they were separated from nouns based on orthography: a capitalized noun that does not start a sentence is likely to be a proper noun. Then a list of proper nouns was collected and manually analyzed. Finally the sentence-initial words were processed using this list;

   (d) A set of prefixoides, for instance само 'self-', полу 'half-', пост 'post-' etc., is annotated as a compound (COM pos-tag) in SynTagRus and usually stored in a separate node. For example, пол года 'half a year' would form two separate tokens. Such tokens must be concatenated into one token;

   (e) Punctuation marks are preserved in XML at the positions where they occurred in the raw text but they do not have their own token-level XML elements (Iomdin and Sizov, 2009) in SynTagRus. Therefore, we extracted punctuation from XML separately, created nodes for all punctuation symbols and assigned them the PUNCT tag directly.

2.2 UD_Russian-GSD

UD_Russian-GSD contains excerpts from Russian Wikipedia (100K tokens, 5K sentences). Its manual annotation at Google was not done directly in the UD annotation scheme; however, both the UD and the Google scheme are based on Stanford Dependencies and thus fairly close. The first set of transformations was already done by Google researchers.[3] The dependency structures and labels were checked manually. The biggest changes were in the annotations of verb patterns, parentheticals of different kind, appositives and other flat syntactic relations in nominal groups, and, predictably, long-distance relations. Since it is not uncommon that some words and text fragments are missing (as a result of fast web-crawling), we treated such cases as a special type of ellipsis and re-annotated the corresponding clauses almost in full. A few 'second order' dependency relations got simplified in UD, for example, `nmod:gobj` and `nmod:tmod` were converted into `nmod`. In contrast, the semantic subjects in passive constructions (marked by the Instrumental case) were labeled as `obl:agent`.

The treebank was not lemmatized, so lemmas were added and manually checked in subsequent releases. The feature annotation was made more consistent with the Russian National Corpus tagset and UD 2.x guidelines.

[3] Ryan McDonald and Vitaly Nikolaev

2.3 UD_Russian-PUD

The treebank is a part of the Parallel Universal Dependencies (PUD) treebanks created for the CoNLL 2017 shared task on Multilingual Parsing from Raw Text to Universal Dependencies (Zeman et al., 2017).

There are 1000 sentences that are taken from the news domain and from Wikipedia. All the sentences are translations from other languages—mostly English, partially also German, French, Italian and Spanish. The same sentences have been translated to and annotated in 18 languages, hence the name *Parallel* UD. Like with GSD, the manual annotation was provided by Google and conformed to their version of the Stanford dependencies. We then converted it to the UD scheme. The treebank has not been lemmatized until v.2.3.

The entire treebank is labeled as a test set and was used for testing in the shared task. In our experiments we use the treebank as a test set as well.

2.4 UD_Russian-Taiga

UD_Russian-Taiga is a new treebank that focuses on registers not covered by the other corpora: blogs, social media, and poetry (21K tokens, 1.7K sentences). It presents constructions specific for web communication, cf. relations between root and emojis, topic hashtags, and nicknames; nominalizations, infinitive roots, and web-specific kinds of ellipsis occurring frequently in the treebank. Less standard word order such as postposed adjectives, and non-projective constructions are characteristic of poetic texts. Overall, the number of words spelled in non-standard orthography and non-standard grammatical forms is higher than in SynTagRus and other Russian UD corpora. UD_Russian-Taiga was annotated directly in UD 2.x style (specifically, it was preprocessed by UDPipe (Straka and Straková, 2017) trained on SynTagRus, then manually corrected). At the time of annotation, the three other Russian treebanks were available in UD version 2.1, so the developers of Taiga could query them for annotation of specific constructions and lexemes and make consistent decisions.

2.5 Comparison of the Treebanks

As the four treebanks come from different sources and were originally created by different teams, there are differences in how they interpret the UD guidelines. We list some of the more significant differences in this section. This is meant as a warning to the users of UD 2.2 and earlier releases, it may also help with understanding the results of the 2017 and 2018 CoNLL shared tasks. We intend to contribute better harmonized versions of the treebanks, which will hopefully become part of the UD 2.3 release in November 2018.

- There are no lemmas in PUD, and the lemmas in GSD are all-uppercase (thus not compatible with SynTagRus and Taiga). Both GSD and PUD will be automatically lemmatized with a model compatible with SynTagRus and Taiga.
- There are discrepancies in how the morphological features are used (see the Notes column in Table 1 for details).
- There are many different auxiliary verbs in GSD (which correspond to auxiliaries in English UD). The only approved auxiliary verb is быть (to be).

Feature	Values	Notes
Gender	Fem Masc Neut	
Animacy	Anim Inan	
Number	Plur Sing	
Case	Acc Dat Gen Ins Loc Nom Par Voc	No Par and Voc in PUD
Degree	Cmp Pos Sup	No Degree in PUD, no Sup in Taiga
Polarity	Neg	
Variant	Long Short	Long appears only in PUD
VerbForm	Conv Fin Inf Part	Inconsistent in PUD
Mood	Cnd Imp Ind	No Cnd in GSD and PUD
Aspect	Imp Perf	
Tense	Fut Past Pres	
Voice	Act Mid Pass	No Mid in PUD
PronType	Prs	Only in PUD and only this value
NumType	Card	Only in GSD and only this value
Reflex	Yes	Only in GSD and PUD
Person	1 2 3	
Gender[psor]	Fem Masc	Only in PUD
Number[psor]	Plur Sing	Only in PUD
Abbr	Yes	Only in Taiga
Foreign	Yes	Not in GSD, insufficient in PUD

Table 1: Features and their values in the Russian treebanks as of UD release 2.2.

- Some language-specific extensions of relation types used in PUD reflect distinctions made in the Google Stanford Dependencies, but they are unimportant in the context of Russian UD and should be reduced to the universal relation label: cc:preconj, nmod:gmod, nmod:poss, obl:tmod.

- In contrast, some language-specific relations are present in the other treebanks but missing from PUD: nummod:entity, nummod:gov, flat:foreign, flat:name. For all four there are sentences in the treebank where they could and should be used.

In summary, GSD bears some similarity with PUD (both converted from the same source annotation style) and Taiga to SynTagRus (maintained by the same people, although not of same origin: SynTagRus has its own source annotation while Taiga is natively UD). GSD has been harmonized with SynTagRus to a higher degree than PUD, effectively making PUD an outlier in many aspects. This can be probably attributed to the fact that PUD entered the UD family later and there have been no harmonization efforts beyond the initial quick-fix conversion, done hastily before the deadline for the CoNLL 2017 Shared Task.

3 Parsing Experiments

To assess the discrepancies among the four UD Russian treebanks we conducted parsing experiments that were previously described by Martínez Alonso and Zeman (2016). Martínez Alonso and Zeman describe the conversion of the Catalan and Spanish AnCora treebanks

to the Universal Dependencies formalism and assess the quality of the resulting treebanks by conducting a series of monolingual, cross-lingual and cross-domain parsing experiments. Since the four Russian treebanks contain not only different genres, but also were created in principally different ways, we decided to re-use the idea of the monolingual cross-domain parsing experiment, where two models were trained for the two UD Spanish treebanks and attachment accuracies were measured using one Spanish treebank to parse the other. The idea behind this experiment is as follows. If the two treebanks were as similar as possible, the differences in parsing accuracy would be due to dataset size and domain change, and not to differences in dependency convention.

We believe that not only does this approach help to reveal annotation errors and dissimilarities, but also highlights the variety of equivalent parses.

3.1 Design

Like Martínez Alonso and Zeman, for all parsing experiments we use a single parser not to obtain state-of-the-art scores, but rather to assess the relative consistency of the Russian UD treebanks of different genres and origins. For the purpose of the parsing experiments we chose the neural graph-based Stanford parser (Dozat et al., 2017), the winner system from the CoNLL 2017 Shared Task (Zeman et al., 2017). We trained models on UD_Russian-SynTagRus and UD_Russian-GSD using the hyperparameters that were used in the CoNLL 2017 Shared Task. Due to their size, UD_Russian-Taiga and UD_Russian-PUD were used only to measure attachment accuracy. We trained one model on GSD and two models on SynTagRus. The first SynTagRus model was trained on the standard training set. The training set for the second model was made by reducing the standard SynTagRus training data to the size of the GSD training set (10% of the full SynTagRus training set). The reduced model was used to show the influence of the size of the data. Then we measured labeled attachment scores on the four parsed test sets and identified an exemplary model – the model that yields the highest scores on the four test files. Finally, for the outputs of the exemplary model, we examined pairs of parsed and gold standard files for every treebank and manually corrected the discrepancies.

Additionally, considering the variety of genres presented in the four corpora, we calculated statistics on out-of-vocabulary words[4] for every test set to ensure that the LAS scores are not affected by genre-specific or topic-specific vocabulary, slang or terminology.

3.2 Results

Table 2 shows labeled attachment scores measured on the four Russian treebanks. No model is decisively the best, but Table 2 clearly demonstrates the benefits of the size of the full SynTagRus training set. There is a great drop in performance when substituting the standard model with the reduced one. The domain change is probably not the main reason for the performance drop. The *Fixed* row shows LAS scores on manually corrected gold standard files. After manual correction, the full SynTagRus model performs better on corrected GSD/PUD test sets than the GSD model, although these test sets are out-of-domain for SynTagRus, and in-domain for GSD.

Table3 shows Content-word Labeled Attachment Score (CLAS), a parsing metric that consid-

[4]By out-of-vocabulary words we mean the words that are present in a test set, but not in the training set.

ers only relations between content words, for the model trained on the full standard training set.

Table 4 shows statistics on out-of-vocabulary words calculated by word form and by lemma[5]. The numbers show that each test set has almost the same number of unknown words. On top of that, we analyzed the top frequent unknown words for each test set and discovered that the unknown words are either different symbols for punctuation marks or proper names. Thereby the results confirm that the LAS scores are not affected by vocabulary.

Model \ Test	SynTagRus	Taiga	GSD	PUD
SynTagRus	**92.14%**	**69.55%**	65.01%	76.73%
– Fixed	–	*70.90%*	*87.41%*	*79.04%*
SynTagRus 10%	86.60%	65.35%	63.57%	74.71%
GSD	67.16%	60.29%	**81.84%**	**78.29%**

Table 2: LAS for UD_Russian-SynTagRus full, UD_Russian-SynTagRus 10% and UD_Russian-GSD models; Fixed: LAS after fixing errors in the test data.

Model \ Test	SynTagRus	Taiga	GSD	PUD
SynTagRus	90.61%	69.17%	65.57%	76.97%
– Fixed	-	70.40%	87.15%	79.49%

Table 3: CLAS for UD_Russian-SynTagRus full model.

Parameter	SynTagRus	Taiga	GSD	PUD
word form	30.18%	30.07%	30.83%	24.73%
lemma	20.49%	22.01%	**26.10%**	21.98%

Table 4: Statistics on out-of-vocabulary tokens (word forms and lemmas) in the testing sets.

3.3 Analysis

This section focuses on the mismatches in the dependency annotation of the gold standard Taiga (gold) and Taiga parsed by SynTagRus full model (predicted). The test and train treebanks represent two poles of the annotation strategy (conversion from another dependency representation vs. annotation from scratch).

It is interesting to compare the annotation consistency of the corpora in terms of functional and content relations. While the SynTagRus model fails to predict the functional relations only 5% of the time in Taiga (cf. 30% on all relations), labeling of `aux` relations is still poor (14%) due to the fact that the subjunctive marker бы 'would' is not treated as auxiliary in SynTagRus. Similarly, a number of words do not fall under the category of `case`, `det`, and `cc` (eg. как 'as', всякий 'any'; also in specific patterns: '–' in *2–4* '2 to 4', то... то... 'now...

[5]UD_Russian-PUD did not contain lemmas in UD 2.2. Therefore we added lemmas using the part of the pipeline(Mediankin and Droganova, 2016) designed specifically for Russian and manually checked them.

then...'). On the contrary, errors in `mark` and `cop` are due to the parser's failure to recognize the structure and usually more distant head.

Of content relations, the cross-parsing model misclassifies 32% inter-clausal, `conj`, and `parataxis` relations; 25% verb arguments; and 25% NP arguments. We can assume that the frequent errors in `parataxis` (55%) are due to complicated and irregular structures of the sentences; however, in some cases, it is a consequence of differing strategies in the annotation of different kinds of parenthesis (cf. `parataxis` vs. `appos` vs. `discourse`). Besides that, there are also genre-specific paratactic patterns in Taiga not seen or underrepresented in SynTagRus such as numbered and bulleted lists, or emoticons that follow or precede the utterances. As for typical verb-argument relations, corpora are inconsistent in the annotation of the 2nd and 3rd arguments, as expected, and in labeling depictive constructions (`xcomp`), whereas the high rate of unlabeled attachment (UA) errors in `nsubj` is an indicator that the model performs poorly on verb-less sentences (underrepresented in SynTagRus). Moreover, errors in `nmod` labeling reveal two problematic constructions: nouns with infinitive and bare Dative complement. Both SynTagRus and Taiga seem to be consistent in their annotation, but in SynTagRus only one noun (смысл 'sense') shows up with infinitives, and Dative cannot be seen without a preposition (cf. Флаг тебе.Dat в руки 'do just as you like', lit. 'flag to you.Dat in hands' typical of net communication genres in Taiga).

Finally, the relations `fixed` and `flat` are not predicted in 40% and 47% cases, respectively. We can see that the list of multi-word expressions is larger in Taiga, as it was updated during the manual annotation (cf. the conjunction не только 'not only', parenthetic в принципе 'in principle', preposition что до 'as for', among others). In Taiga, `flat` is applied to words repeated two or more times (e.g. всех всех 'all', не не не 'no no no'; also in the idiomatic construction уж кто кто, а... 'I don't know about anyone else, but...'), the pattern which the SynTagRus model fails to parse the same way. The error rate for punctuation is considerably low (2%) compared to those for flat relations, but analysis shows a difference in the attachment of terminal symbols and commas in nested clauses in train and test corpora.

It should not be forgotten, however, that the parser's errors can be caused not only by the inconsistency in corpus annotation per se but also by different distribution of text genres, syntactic structures, and lexicon (cf. abundance of abbreviated words and misspellings in Taiga not presented SynTagRus). The classification of cross-parsing errors by relation types suggests some heuristics to distinguish annotation inconsistency among other reasons and reveal the patterns tagged differently in different corpora. If we consider the ratio of UA errors and LA errors (i.e. cases in which the arcs are attached correctly but the labels are predicted incorrectly), then numbers close to zero usually signal inconsistency in annotation and mismatches in lexical inventories. The predominance of UA errors over LA errors may suggest that the target construction is unlikely to have been present in the training corpus. Nevertheless, in the case of the flat relation `fixed`, the mismatch in patterns covered by the annotation guidelines leads to the parser's inability to guess the unlabeled structure and thus the ratio of UA to LA errors is high.

The limitation of the method is that the UA/LA error ratio seems to be different for different relations and depends on the total LAS score in a particular pair of train and test treebanks. The complexity of parsed structures and the degree of overlap of lexical units labeled by a particular relation may also affect the UA/LA ratio. Overall, more work should be done to estimate the impact of the training and test corpus size on the findings.

Last but not the least, analysis of cross-parsing mismatches helps to reveal a set of ambiguous patterns in the treebank, i.e. those that can be parsed in different ways, sometimes with certain difference in interpretation. As an example, cf. the label `ccomp` (gold) and `acl` (parsed) on the verb платили 'pay' in главное[head], чтобы платили[ccomp vs. acl], и рейтинг рос... 'the main thing[head] is to be paid[ccomp vs. acl], and to get a higher rating...'.

4 Future Work

According to our observations the following changes need to be made to improve consistency across the four treebanks:

- fix punctuation attachment; revise the `parataxis` vs `ccomp` distinction in reported speech patterns; revise the annotation of adverbs in UD_Russian-SynTagRus;
- revise the annotation of auxiliary verbs; revise lemmatization; revise the morphological features in UD_Russian-GSD;
- revise the annotation of a number of specific patterns in UD_Russian-Taiga;
- add lemmas; revise language-specific syntactic relations; revise the morphological features in UD_Russian-PUD.

5 Conclusion

We have presented Russian UD treebanks: UD_Russian-SynTagRus, UD_Russian-GSD, UD_Russian-Taiga, and UD_Russian-PUD and described a series of experiments aimed at checking syntactic annotation consistency within the four treebanks. We have extended the method previously proposed by Martínez Alonso and Zeman (2016) to confirm that the LAS scores are not affected by genre-specific or topic-specific vocabulary. We have presented the LAS scores which can serve as criteria for deciding on the optimal corpus mix for parsing experiments. The analysis of mismatches in the test and predicted relations reveals a list of patterns that could be annotated more consistently in the four corpora.

Acknowledgments

The work by Kira Droganova and Daniel Zeman was partially supported by the GA UK grant 794417, the SVV project number 260 453, the grant 15-10472S of the Czech Science Foundation (GAČR), and FP7-ICT-2009-4-249119 (MŠMT 7E11040). Research by Olga Lyashevskaya was done within the framework of the Academic Fund Program at the National Research University Higher School of Economics (HSE) in 2018 (grant 18-05-0047) and by the Russian Academic Excellence Project «5-100».

References

Alzetta, C., Dell'Orletta, F., Montemagni, S., and Venturi, G. (2018). Dangerous relations in dependency treebanks. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories (TLT16)*, pages 201–210, Praha, Czechia.

Boguslavsky, I., Iomdin, L., Timoshenko, S., and Frolova, T. (2009). Development of the russian tagged corpus with lexical and functional annotation. In *Metalanguage and Encoding Scheme Design for Digital Lexicography. MONDILEX Third Open Workshop. Proceedings. Bratislava, Slovakia*, pages 83–90.

Boyd, A., Dickinson, M., and Meurers, W. D. (2008). On detecting errors in dependency treebanks. *Research on Language and Computation*, 6(2):113–137.

Brants, T. (1997). The negra export format for annotated corpora. *University of Saarbrücken, Germany*.

Brants, T. and Skut, W. (1998). Automation of treebank annotation. In *Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning*, pages 49–57. Association for Computational Linguistics.

de Marneffe, M.-C., Grioni, M., Kanerva, J., and Ginter, F. (2017). Assessing the annotation consistency of the universal dependencies corpora. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 108–115.

De Smedt, K., Rosén, V., and Meurer, P. (2015). Studying consistency in ud treebanks with iness-search. In *Proceedings of the Fourteenth Workshop on Treebanks and Linguistic Theories (TLT14)*, pages 258–267.

Dickinson, M. and Meurers, W. D. (2003). Detecting inconsistencies in treebanks. In *Proceedings of TLT*, volume 3, pages 45–56.

Dozat, T., Qi, P., and Manning, C. D. (2017). Stanford's graph-based neural dependency parser at the conll 2017 shared task. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30.

Dyachenko, P., Iomdin, L., Lazursky, A., Mityushin, L., Podlesskaya, O., Sizov, V., Frolova, T., and Tsinman, L. (2015). Sovremennoe sostoyanie gluboko annotirovannogo korpusa tekstov russkogo yazyka (syntagrus). *Trudy Instituta Russkogo Yazyka im. V. V. Vinogradova*, (6):272–300.

Iomdin, L. and Sizov, V. (2009). Structure editor: a powerful environment for tagged corpora. *Research Infrastructure for Digital Lexicography*, page 1.

Kaljurand, K. (2004). Checking treebank consistency to find annotation errors. Technical report at ResearchGate, `https://www.researchgate.net/publication/265628472_Checking_treebank_consistency_to_find_annotation_errors`.

Kulick, S., Bies, A., Mott, J., Maamouri, M., Santorini, B., and Kroch, A. (2013). Using derivation trees for informative treebank inter-annotator agreement evaluation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 550–555.

Martínez Alonso, H. and Zeman, D. (2016). Universal dependencies for the ancora treebanks. *Procesamiento del Lenguaje Natural*, (57):91–98.

Mediankin, N. and Droganova, K. (2016). Building NLP pipeline for russian with a handful of linguistic knowledge. In Chernyak, E., Ilvovsky, D., Skorinkin, D., and Vybornova, A., editors, *Proceedings of the Workshop on Computational Linguistics and Language Science*, pages 48–56, Aachen, Germany. NRU HSE, CEUR-WS.

Mel'čuk, I. A. (1981). Meaning-text models: a recent trend in soviet linguistics. *Annual Review of Anthropology*, 10(1):27–62.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association.

Straka, M. and Straková, J. (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, Vancouver, Canada. Association for Computational Linguistics.

Zeman, D., Popel, M., Straka, M., Hajič, J., Nivre, J., Ginter, F., Luotolahti, J., Pyysalo, S., Petrov, S., Potthast, M., Tyers, F., Badmaeva, E., Gökırmak, M., Nedoluzhko, A., Cinková, S., Hajič jr., J., Hlaváčová, J., Kettnerová, V., Urešová, Z., Kanerva, J., Ojala, S., Missilä, A., Manning, C., Schuster, S., Reddy, S., Taji, D., Habash, N., Leung, H., de Marneffe, M.-C., Sanguinetti, M., Simi, M., Kanayama, H., de Paiva, V., Droganova, K., Martínez Alonso, H., Çöltekin, Ç., Sulubacak, U., Uszkoreit, H., Macketanz, V., Burchardt, A., Harris, K., Marheinecke, K., Rehm, G., Kayadelen, T., Attia, M., Elkahky, A., Yu, Z., Pitler, E., Lertpradit, S., Mandl, M., Kirchner, J., Fernandez Alcalde, H., Strnadova, J., Banerjee, E., Manurung, R., Stella, A., Shimada, A., Kwak, S., Mendonça, G., Lando, T., Nitisaroj, R., and Li, J. (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.

Measuring Implemented Grammar Evolution

Dan Flickinger

Stanford University

danf@stanford.edu

ABSTRACT

Natural language grammar implementations that are constructed manually can grow over time to become large, complex resources covering an ever wider range of language phenomena. It would be useful both for the grammarian and for the users of a grammar to have a good understanding of how a later version of that grammar behaves in contrast to an earlier version, particularly in terms of the treatment of linguistic phenomena. This paper presents a case study of the evolution of an implemented grammar, comparing two versions of the same grammar by measuring several properties of the analyses recorded using that grammar in two corresponding versions of an associated dynamic treebank.

KEYWORDS: implemented grammar evolution, linguistic phenomena, dynamic treebank.

Introduction

Manually constructed grammars of natural languages can grow over time to become large, complex resources that provide detailed linguistic analyses of a wide range of language phenomena. It would be useful both for the grammarian and for the users of a grammar to have a good understanding of how a later version of that grammar behaves in contrast to an earlier version, particularly in terms of the treatment of linguistic phenomena. For the grammarian, such a comparison would show changes in the breadth and depth of analyses of phenomena as a measure of progress, and for the user, the comparison would help set expectations about how suitable the grammar will be for a specific corpus or genre of text. Treebanks can play a central role in measuring the evolution of a grammar, by enabling fine-grained comparison of the grammar's changes in precision and recall over a constant corpus, not only for whole sentences but also for phenomena identified in the analyses of these sentences recorded in the treebanks. This paper presents a case study of grammar evolution, comparing two versions of the same grammar, the English Resource Grammar (Flickinger, 2000, 2011), by measuring several properties of the analyses recorded using that grammar in two corresponding versions of the Redwoods Treebank (Oepen et al., 2004).

1 Grammars and Treebanks

The English Resource Grammar (ERG) is an open-source, broad-coverage, declarative grammar implementation for English, developed within the Head-driven Phrase Structure Grammar (HPSG) framework of Pollard and Sag (1994). This linguistic framework places most of the burden of linguistic description on the lexicon, employing a relatively small number of highly schematic syntactic rules to combine words or phrases to form larger constituents. Each word or phrase (more generally, each sign) is defined in terms of feature structures where the values of attributes are governed by general principles such as the Head Feature Convention, which ensures the identity of a particular subset of attribute-value pairs on a phrasal node and on one distinguished daughter, the head of the phrase. Many of these generalizations aim to be language-independent, constraining the space of possible analyses for linguistic phenomena. Central to the HPSG framework is the inclusion in each sign of constraints on multiple dimensions of representation, including at least syntactic and semantic properties, so that rules, lexical entries, and principles determine semantic well-formedness just as they do syntax. Under continuous development at CSLI since 1994, the ERG claims to provide syntactic and semantic analyses for the large majority of common constructions in written English text.

The two versions of the grammar used for comparison here are the most recent stable version of the grammar (ERG 2018) and the previous stable version from four years earlier (ERG 1214). The 2018 version consists of a 40,000-word lexicon instantiating 1400 leaf types in a large lexical type hierarchy, as well as 110 derivational, inflectional, and punctuation rules, and 250 syntactic rules. At this level of description, the previous 1214 version is somewhat smaller, consisting of a 38,000 word lexicon, 1250 leaf types, 80 lexical rules, and 210 syntactic rules.

Associated with each stable version of the ERG is a corresponding treebank which is manually updated to validate and improve the recorded analysis of each sentence in the treebank that can be correctly parsed using the grammar. For each sentence, an annotator has used a customized tool to identify the intended analysis by making choices from among the set of binary *discriminants*, which encode contrasts among alternative analyses at lexical and phrasal levels (Carter, 1997). These sentence-specific discriminants can be automatically reapplied to a freshly produced parse forest that resulted from using a revised version of the grammar, so that

the human annotator need only attend to those sentences in the treebank where the existing discriminant choices did not fully resolve the ambiguities in the new parse forest. This need for additional annotation can result from additional ambiguity introduced by the revised grammar, from a change of analysis such that the grammar no longer makes available the previously recorded analysis, or from newly available analyses for sentences which were previously not in the scope of the grammar. Where the set of available analyses has changed for a sentence in the treebanked corpus, the annotator typically only needs to make a small number of additional decisions among the newly presented discriminants, apart from previously unparsable sentences, and hence the annotation costs of keeping this dynamic treebank up to date with each new stable grammar version remains manageable.

The two versions of the ERG-parsed treebank used for this comparison are subsets of the Redwoods Ninth Growth, parsed with ERG 1214, and the emerging Tenth Growth, parsed with ERG 2018. Both treebank versions record annotations for the same set of varied text corpora, consisting of 58,000 sentences from the following sources:

- Eric Raymond essay on open-source software (769 items)
- Wikipedia – 100 articles from Wikipedia on computational linguistics (11556 items)
- SemCor – Subset of sense-annotated portion of Brown corpus (3103 items)
- DeepBank – Wall Street Journal text in the Penn Treebank, secs. 00–04 (9648 items)
- Verbmobil – Dialogues on meeting scheduling and hotel booking (12393 items)
- LOGON – Brochures and web text on Norwegian tourism (11596 items)
- Tanaka – Subset of English learner data used for Pacling 2001 (3000 items)
- E-commerce – Customer emails on product sales (5793 items)

One important difference in the two versions of the treebank is that two distinct annotation tools were used, both employing discriminant-based disambiguation, with the earlier 'classic' one enabling choice only among the top-ranked 500 parses for each sentence (based on a previously trained model), but the later one preserving the full (packed) parse forest. This difference has two primary effects: first, for at least some items in the corpus, the best analysis was not ranked among the top 500, and thus was not available to the annotator using the classic tool for the Ninth Growth; and second, preserving the packed full forest using the latter tool eliminated the sometimes considerable processing cost of unpacking, so that more complex sentences could be parsed within reasonable resource limits and presented to the annotator for disambiguation for the Tenth Growth. As a result of the reduction in parsing cost for the full-forest method, the raw coverage numbers for the two versions of the treebank are not directly comparable, although the differences can be mitigated by noting for each version the number of items in each corpus for which parse failure was affected by resource limitations.

2 Measurements of Evolution

We have already seen that the two versions of the grammar differ noticeably in the inventories of linguistic objects that comprise them, with the more recent 2018 version containing a slightly larger lexicon, and markedly more rules, both lexical and phrasal. To study the effects of these changes in the parsing of text, it should be worthwhile to quantify both coarse-grained properties such as overall coverage (parsability) and percentage of items where the recorded analysis has changed, as well as more fine-grained properties such as the usage of particular rules and lexical types, as indicators of the frequency of linguistic phenomena in the corpus that are within the scope of the grammar.

2.1 Corpus-level metrics

Table 1 provides a high-level view of the coverage of the 1214 version of the grammar for each component corpus, with the data taken from the file `redwoods.xls` included with the ERG source files for this earlier version. The table reports for each component the total number of items, the average number of tokens per item, the number (and percentage) of items parsed, and the items verified and thus included in the treebank.

Corpus	Items	Tokens	Parsed	Verified
Essay	769	22	711 (92%)	604 (79%)
Wikipedia	11558	18	10649 (92%)	9237 (80%)
SemCor	3103	18	2923 (94%)	2560 (83%)
DeepBank	9648	21	9255 (96%)	8450 (88%)
Verbmobil	12393	7	11949 (96%)	11406 (92%)
LOGON	11956	14	11664 (98%)	11024 (92%)
Tanaka	3000	12	2890 (96%)	2814 (94%)
E-commerce	5793	9	5627 (97%)	5420 (95%)

Table 1: Redwoods Ninth Growth (ERG 1214)

As noted above, the direct comparison of coverage numbers for the two versions of the Redwoods treebank is not fully informative, especially because of the differing effects of resource limits when parsing to prepare the data for annotation. However, the number of items affected by resource limits was typically different by only one or two percent for each component corpus, so that for example the Brown sub-corpus of 3103 items saw 55 unparsed items hitting resource limits in the Ninth Growth contrasted with 17 in the Tenth Growth. Hence the comparison of verified coverage annotations for the two versions of the treebank shown in Table 2 should be viewed with this potential offset in mind, varying slightly from component to component.

Corpus	Ninth (%)	Tenth (%)
Essay	79	93
Wikipedia	80	89
SemCor	83	91
DeepBank	88	93
Verbmobil	92	93
LOGON	92	96
Tanaka	94	96
E-commerce	95	98

Table 2: Verified coverage for Redwoods with ERG 1214 vs. ERG 2018

Setting aside these minor effects from differences in the treebanking tools, the improvements in the number of successfully annotated items when comparing the Ninth and Tenth Growths should correlate with improvements in the linguistic analyses introduced during the four years of development between the 1214 and 2018 versions of the ERG. An examination of some of these changes in the grammar's treatment of linguistic phenomena is the focus of the next section.

2.2 Phenomenon-based metrics

For component corpora where the average number of tokens per sentence is lower, such as for the scheduling dialogues of Verbmobil or the English learner sentences of Tanaka, the change in the number of successfully annotated items is not dramatic, suggesting that enhancements in the depth or breadth of linguistic analysis are less likely to be evidenced in these sentences. In sharp contrast, those corpora with sentences of greater average length, such as the Brown corpus of SemCor and the newspaper text of DeepBank, should help to illuminate substantive changes in the grammar from the older version to the newer one.

One example of a familiar linguistic phenomenon that might be expected to remain steady in its frequency of use within the two versions of the treebank is the passive, whose implementation in the ERG is divided into several subtypes, including not only the ordinary structure in *The cat was chased by the dog* but also the more interesting variants in *That author is looked up to by everyone* and *She can be relied on to succeed*. However, Table 3 shows several nontrivial changes from one version of the treebank to the next, including a notably higher number of uses of the ordinary passive affecting the direct object of the verb, and of the *relied on* type. In addition, the table reflects the addition in ERG 2018 of an analysis of infrequently used passives of the *looked up to* variety, lacking in the 1214 version of the grammar.

Passives	Ninth	Tenth
All types	10786	11728
was admired	10151	11039
was given (something)	553	585
was relied on	309	403
was believed that S	73	80
was looked up to	0	3

Table 3: Number of items with passive in two versions of Redwoods

The increase in the presence of the most frequent type of passive appears to be correlated with the higher levels of annotation success for those corpora with a greater average sentence length. For example, the 769-sentence open-source essay improved by 14% for successfully annotated sentences, and the number of annotated sentences in that essay using the ordinary passive increased from 137 to 174. Similarly, successful annotation for the 3100-sentence subset of the Brown corpus (SemCor) improved from 83% to 91%, with a corresponding increase in the use of the ordinary passive from 494 items to 583. This alignment of increased use of passives with increased verified coverage does not provide any clear indication that improvements in the grammar's analysis of passives have contributed materially to greater parsing success, since it may well be the case that other grammar improvements enabled more sentences to be parsed, and those sentences simply also make use of passives, so they appear in greater numbers in the recorded analyses.

Another construction type that appears frequently across genres is the relative clause, which is again analyzed via several subtypes in the ERG, as shown in Table 4. Here, too, the overall frequency of use of relative clauses appears to be little affected by changes in the grammar from the 1214 version to 2018, with the greater number of uses in the Tenth Growth roughly correlated with overall improvements in successful annotation. This consistency across versions holds as well for phenomena also grouped with relative clauses by (Huddleston and Pullum, 2002) in their chapter 12, including free relatives (*whatever we needed*) and *it*-cleft constructions

(*it was on Tuesday that we scheduled the workshop*). Thus, while the 2018 version of the grammar draws some finer distinctions in its analysis of relative clauses than did the 1214 version, for example distinguishing PP fillers (*on whom we relied*) from NP fillers (*who we admired*), the differences are not reflected in dramatic changes in the number of analyzed sentences exhibiting this class of phenomena.

Relative clauses	Ninth	Tenth
All types	8205	8870
the book which we admired	4223	4722
the book admired by everyone	2995	3304
the book we admired	1021	1045
the guy to talk to	643	700
the day we arrived	88	100

Table 4: Number of items with relative clause in two versions of Redwoods

In contrast to these two examples of frequently occurring phenomena where inspection of the treebanks suggests that little has changed in the analyses provided by the two versions of the grammar, there are phenomena whose analysis is clearly different in the two grammar versions. One relatively frequent example is found in constrained but relatively productive noun-noun compounds where the left member is inflected for plural, as in *systems analyst* or *weapons production*, contrasted with **flowers garden* or **towels rack*. These plural compounds also include conjoined nouns as left members, as in *health and family welfare agencies*. The 1214 version of the ERG, presumably sensitive to the ungrammaticality of compounds such as **flowers garden*, did not provide a productive compounding rule for plural or conjoined left members, instead attempting to lexically list frequently found non-heads such as *systems* or *weapons*. In the 2018 version of the grammar, syntactic constructions have been added to admit plural compounds, improving overall coverage at the cost of overgenerating previously rejected compounds such as **flowers garden*. The Tenth Growth records 1055 items whose analyses use these constructions, and many if not most of those sentences would have not been included in the Ninth Growth, apart from the ones for which a use-specific lexical entry had been included in the lexicon.

Several other syntactic constructions have been added for the 2018 version of the grammar, and while they are not individually highly frequent, their successful use helps in the aggregate to account for some of the increases in annotations for the Tenth Growth compared to the Ninth. Table 5 shows the number of items analyzed in the Tenth Growth using some of these new constructions, where these items lack correct analyses in the Ninth Growth.

Other phenomena	Example	Tenth
Appositive with measure-NP	*Summit hike, 20 km*	65
Indef NP as clause modifier	*A good scholar, it was likely she would thrive.*	44
Coordination of selected-for PPs	*relied on us and on you*	26
Parenthetical adjective	*the parent (subsuming) class*	24

Table 5: Number of items using constructions only included in ERG 2018

3 Related work

Since the discriminant-based approach to treebank construction and maintenance employed for Redwoods has also been adopted by developers of the TREPIL project for treebanks recording

analyses using grammars in the Lexical-Functional Grammar linguistic framework, the concept of grammar and treebank evolving in lockstep is also explored there (Rosén et al., 2005, 2016). However, work in this project has not to date focused on using distinct successive stages of the grammar/treebank pair to help to illuminate systematic changes made to the grammar from one version to the next.

A more direct connection to the present work is a hypothesis proposed in Bender et al. (2012), suggesting on p. 192 that software developed for searching treebanks could be adapted to "facilitate the exploration of the evolution of analyses either of particular examples in an implemented grammar, or of classes of examples." The authors acknowledge that such an extension to their search interface would require considerable additional effort; the present study has not aimed at designing such a user interface, instead employing direct textual search within the recorded treebank analyses for the names of specific constructions or groups of constructions associated with phenomena of interest. This direct search method demands a level of familiarity with the naming conventions used for rules and lexical types in the ERG, but these are documented at http://moin.delph-in.net/ErgTop.

It should be noted that the term *grammar evolution* is sometimes used to refer to the process of grammar change over time within a linguistic community, but this is not related to the present study, where the language being analyzed is presumed to be constant for the moment, and it is instead the grammar's *implementation* which is treated as evolving over time in order to express improved analyses of the language.

4 Conclusion and Outlook

Dynamic treebanks such as Redwoods or those developed in the TREPIL project have multiple uses; this study is an exploration of one additional use of such annotations, to provide insights about the often murky nature of the changes made to a manually constructed grammar over relatively long periods of time. By examining frequencies of the use of specific rules or groups of rules used in the analysis of individual linguistic phenomena, an observer can obtain a better understanding of what has changed from one version of the grammar to the next, to help in explaining more readily observed changes in coverage of the grammar when applied to a corpus. For a more complete understanding of the state of the grammar, the implementation should be accompanied by a rich inventory of linguistic phenomena that the grammarian aspires to analyze, perhaps anchored in a comprehensive pencil-and-paper grammar such as (for English) the Cambridge Grammar of the English Language (Huddleston and Pullum, 2002). Documenting this inventory and the mapping from the English Resource Grammar to such a resource should be a priority for further work on this approach.

Another clearly desirable improvement over the methods described here would be to enable searches of the treebank for specific phenomena without requiring explicit and sometimes tedious mention of each rule involved by name, though this would require the definition of a nontrivial mapping between an inventory of linguistic phenomena at varying levels of abstraction, and the specific constructs defined in the grammar. A small beginning in this direction can be found in the Redwoods Tenth Growth with analyses of almost all of the example sentences cited by (Huddleston and Pullum, 2002) in their chapter 12 on relative clauses, a resource also used by (Letcher, 2018) in a distinct and promising approach to phenomenon discovery.

Acknowledgments

Part of this work was conducted during a fellowship at the Oslo Center for Advanced Study at the Norwegian Academy of Science and Letters, and I am grateful for the excellent research environment and support that CAS provided.

References

Bender, E. M., Ghodke, S., Baldwin, T., and Dridan, R. (2012). From database to treebank: On enhancing Hypertext Grammars with grammar engineering and treebank search. *Language Documentation & Conservation Special Publication No. 4 Electronic Grammaticography*, pages 179–206.

Carter, D. (1997). The TreeBanker. A tool for supervised training of parsed corpora. In *Proc. of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering*, pages 9–15, Madrid, Spain.

Flickinger, D. (2000). On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(01):15–28.

Flickinger, D. (2011). Accuracy v. Robustness in grammar engineering. In Bender, E. M. and Arnold, J. E., editors, *Language from a Cognitive Perspective: Grammar, Usage and Processing*, pages 31–50. CSLI Publications, Stanford, CA, USA.

Huddleston, R. and Pullum, G. K. (2002). *The Cambridge Grammar of the English Language*. Cambridge University Press, Cambridge, UK.

Letcher, N. (2018). *Discovering syntactic phenomena with and within precision grammars*. PhD thesis, University of Melbourne.

Oepen, S., Flickinger, D., Toutanova, K., and Manning, C. D. (2004). LinGO Redwoods. A rich and dynamic treebank for HPSG. *Research on Language and Computation*, 2(4):575–596.

Pollard, C. and Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. The University of Chicago Press, Chicago, IL, USA.

Rosén, V., Meurer, P., and de Smedt, K. (2005). Constructing a parsed corpus with a large LFG grammar. In Butt, M. and King, T. H., editors, *Proceedings of the LFG '05 Conference, University of Bergen*, Stanford, CA, USA. CSLI Publications.

Rosén, V., Thunes, M., Haugereid, P., Losnegaard, G. S., Dyvik, H. J. J., Samdal, G. I. L., Meurer, P., and de Smedt, K. (2016). The enrichment of lexical resources through incremental parsebanking. *Language Resources and Evaluation*, 50(2):291–319.

Defining Verbal Synonyms: between Syntax and Semantics

Zdeňka Urešová, Eva Fučíková, Jan Hajič, Eva Hajičová

Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics
Prague, Czech Republic

{uresova,fucikova,hajic,hajicova}@ufal.mff.cuni.cz

ABSTRACT

While studying verbal synonymy, we have investigated the relation between syntax and semantics in hope that the exploration of this relationship will help us to get more insight into the question of synonymy as the relationship relating (similar) meanings between different lexemes. Most synonym lexicons (Wordnets and similar thesauri) are based on an intuition about the similarity of word meanings, or on notions like "semantic roles." In some cases, syntax is also taken into account, but we have found no annotation and/or evaluation experiment to see how strongly can syntax contribute to synonym specification. We have prepared an annotation experiment for which we have used two treebanks (Czech and English) from the Universal Dependencies (UD) set of parallel corpora (PUDs) in order to see how strong correlation exists between syntax and the assignment of verbs in context to pre-determined (bilingual) classes of synonyms. The resulting statistics confirmed that while syntax does support decisions about synonymy, such support is not strong enough and that more semantic criteria are indeed necessary. The results of the annotation will also help to further improve rules and specifications for creating synonymous classes. Moreover, we have collected evidence that the annotation setup that we have used can identify synonym classes to be merged, and the resulting data (which we plan to publish openly) can possibly serve for the evaluation of automatic methods used in this area.

KEYWORDS: Synonyms, lexical resource, parallel corpus, annotation, interannotator agreement, syntax, semantics, universal dependencies, valency.

1 Introduction and motivation

While current NLP systems using modern machine learning methods, such as Deep Learning, minimize the amount of linguistic information necessary to build many successful applications, we believe that semantically-based lexical resources are necessary for linking linguistic content to real world knowledge. At the same time, building such resources – so far often largely manual effort – is time-consuming and could certainly be automated, at least to a certain degree, especially when linguistically annotated data (corpora) are available: "High quality linguistic annotation is the vehicle that has, to a large degree, enabled current successful Natural Language Processing systems." (Palmer et al., 2017).

We study verbal synonymy, as defined in existing lexicons. For the purpose of this paper, we have focused on CzEngClass (Urešová et al., 2018a,c), a bilingual verbal synonym lexicon. In CzEngClass, synonymy is specified by lexical meaning of the synonym class members related to their syntactic behavior by using "contextual constraints," which map semantic roles to valency (and its morphosyntactic realization), to constrain the otherwise intuitive process of deciding which verbs (verb senses) can be considered synonymous. As shown in (Urešová et al., 2018c), the inter-annotator agreement, however, is less than satisfactory, and thus a question arises what else could potentially be used to help the lexicon annotator / creator to decide the synonymy question.

To start with something relatively simple, in this paper we explore the influence of syntax (as realized in the Universal Dependencies (UD)[1] (Nivre et al., 2016) annotation standard) on the decisions made by annotators when assigning a verb in textual context (i.e., in a corpus) to a particular synonym class. We are aware that both the intuition of the annotator about the actual meaning of the verb (in textual context) will certainly play a more important role than a purely syntactic phenomena. One could even argue that especially in a bilingual (or multilingual) context, it is clear from the onset that pure syntax cannot be shared among the synonymous verbs. However, even if it is hard to easily turn around the observation that verbs that fall into the same class exhibit similar syntactic behavior (Levin, 1993) (meant, of course, in the monolingual setting), there could be a correlation that might be explored. In addition, there has not been a situation yet when two or more languages share the same annotation specification, as UD now allows. Thus, if a correlation between (dependency) syntax – namely, between the presence of what UD calls "core dependents" (Zeman, 2017) – and the agreement on assigning the synonym class (made more robust by using bilingual Czech-English material, corresponding to the way CzEngClass is being built) will be found, the findings could possibly be used in e.g., extension of the lexicon with additional (semi-)automatic means. Moreover, it could possibly also bring some insight to the problem of synonymy definition and detection itself. We approach this question by means of a parallel corpus-based verb sense annotation, followed by investigating a correlation between such annotation and the verb's immediate dependency syntax context (as represented by its core dependents). We realize that UD core dependents are only a subset of all possible syntactic features, but we believe that (due to their "coreness", which includes some oblique dependents as well, see Sect. 5.1) they do represent the most of those features that can be reasonably generalized. On the other hand, morphological features of verb dependents are, based on our experience with several past NLP tasks, too language specific to be taken into account in the context of synonymy investigation (even though with more data, it would certainly be appropriate to test them as well).

[1] Universal Dependencies is a project that is developing cross-linguistically consistent treebank annotation for many languages (`http://universaldependencies.org/introduction.html`).

The structure of the paper is as follows. In Sect. 2, the main direction of the paper is motivated by arguing that both semantics and syntax necessarily play an important role in the definition of synonymy, and then the specification of the synonym classes in CzEngClass, which is used as the main dataset for the subsequent experiments, is briefly described. In Sect. 3, the parallel, UD-annotated corpus used in the present experiment is introduced. The annotation part of the experiment is described in Sect. 4, together with basic quantitative evaluation of inter-annotator agreement. Next, we describe annotation post-processing to arrive at a dataset needed for exploring the role of syntax in verb assignment into classes. This exploration and its results are discussed in Sect. 5 and the side effect of the present annotation experiment in Sect. 6. We conclude in Sect. 7, where also some possible next steps and future investigation directions are identified.

2 Synonymy

2.1 Syntax and Semantics in the Definition of Synonymy

Functionally-oriented linguistics understands syntax as an inseparable part of semantics and pragmatics (Wu, 2017). Both within and across languages, there are strong regularities in the relationship between thematic roles and syntactic functions (Bowerman, 2002). Those rules that link thematic roles such as agent and patient to syntactic functions such as subject and direct object are called "linking rules," cf. e.g. (Alishahi and Stevenson, 2010). These correspondences has been extensively studied, e.g., (Levin, 1993; Levin and Hovav, 2005; Kettnerová et al., 2012; Čech et al., 2015), but there are still differences in opinions on the nature of these rules (Hartshorne et al., 2010).

In our opinion, to describe the interplay of syntax and semantic relations in a sentence by linking syntactic (predicate-argument structure) and semantic (semantic roles) phenomena is essential in order to understand the meaning of a sentence. In this context, we believe, in line with (Urešová et al., 2018a), that it is inadequate to analyze just one structure - whether syntactic or semantic, cf. (Pustejovsky, 1991). However, in order to study the relative importance of these two components, we will focus on how much the knowledge of (purely) syntactic structure can help to correctly assign a synonym class to an occurrence of a verb in running text (corpus).

2.2 The CzEngClass synonym lexicon

We are using CzEngClass (Urešová et al., 2018a,c,d) as a lexicon containing verb synonyms. The grouping of verb senses in this lexicon is based on context expressed by a common set of semantic roles assigned to each class and their ability to map these roles to valency arguments of the individual verbs in that class. In other words, this lexicon acknowledges the role of syntax and semantics (in its treatment of synonymy) as described in Sect. 2.1. The lexicon is currently bilingual, meaning that each class contains both Czech and English verbs (synonymous verb senses), presumably mapable into the same ontological concept in both languages. The entries also contain examples and links to external lexical resources in which the referenced verbs are described from various other perspectives (FrameNet (Baker et al., 1998; Fillmore et al., 2003), VerbNet (Palmer, 2009), PropBank (Kingsbury and Palmer, 2002), WordNet (Fellbaum, 1998), and English and Czech valency lexicons (Cinková, 2006; Hajič et al., 2003)). CzEngClass currently contains 200 classes grouping approx. 3500 verb senses. Part of the lexicon (60 classes) has been tested for inter-annotator agreement (Urešová et al., 2018c), which appears to be low by the standard metrics (Cohen's and Fleiss' kappa), but some other metrics developed

specifically for lexicon annotation show more positive figures (Sect. 7.4 of (Urešová et al., 2018c)).

Table 1 illustrates a (simplified) class and the context (semantic role to valency argument mappings) for each member, as taken from (Urešová et al., 2018b). While some entries in

	Roles		
	Agent	Asset_Patient	Harmful_situation
protect	ACT	PAT	EFF
conserve	ACT	PAT	#sth
insulate	ACT	PAT	ORIG
...	...	...	...
chránit	ACT	PAT	EFF
bránit se	ACT	ACT	PAT
zaclonit	ACT	PAT	ADDR
ochraňovat	ACT	PAT	EFF
zastávat se	ACT	PAT	#sb,REG
...	...	...	...

Table 1: Role-to-valency mappings for the PROTECT class

Table 1 show very regular mapping from valency to the class' semantic roles (one English verb, *protect*, and two Czech ones *chránit, ochraňovat* have the same valency - ACT, PAT, EFF), others show that different syntactic structures can have the same semantic roles (i.e., the same meaning), and presumably also vice versa. Please note that the "deep syntactic" valency labels, as opposed to the UD annotation of core dependents, already represent certain "normalization", for example for passivization and other diatheses etc.).

This leads us to the formulation of the main question, whether there is any correlation between the syntactic behavior of the verbs in the synonym classes and the agreement in assigning these classes to verb occurrences in a running text (i.e., when effectively doing the word sense disambiguation task). Should there be a strong correlation, perhaps the synonymy relation for grouping verb senses in classes could be defined in simpler terms - using just dependency syntax instead of the complex semantic-roles-to-valency mapping, as used in CzEngClass.

3 The Corpus and Text Selection for the Experiment

For the annotation experiments described below, we use the Czech and English parallel treebanks ("PUD") from the UD collection, version 2.2 (Nivre et al., 2018). This corpus has been selected since according to (Urešová et al., 2018c) it has not been used in the semi-automatic CzEngClass version 0.2 (Urešová et al., 2018b) creation (avoiding thus the "training data bias"); at the same time, it is an aligned parallel corpus that is annotated for UD dependency syntax, allowing the type of experiment we are aiming at. We have selected aligned pairs of sentences in which at least one verb has been found in at least two of the 60 synonym classes which are fully annotated for synonym class membership in CzEngClass, version 0.2. Such verbs are deemed to have at least two different senses (Urešová et al., 2018a). Together with its aligned verb in the other language they form a pair, and together with the sentences in which they have been found they form an *annotation item* (more precisely, a pair of annotation items).

For each annotation item, classes have been pre-selected based on the verb in question. Each annotation item consists of between two and five possible classes to be assigned by the annotators,

as described below in Sect. 4. However, the verbs in the opposite language have been removed from the set of class members presented to the annotators; i.e., the annotators have not seen the annotation item from the other language. On each side, an "Other" class has been added to allow for annotating cases when no suitable pre-selected class was deemed to be adequate for the meaning of the annotated verb in the context of the sentence in which it occurs.

4 The Annotation Experiment

4.1 The Task

We have hired ten annotators, and divided the annotation items in such a way that each annotation item has been annotated by three of them. As mentioned above, the annotation items have been monolingual (but we did keep the alignment information between the pairs, even though hidden to the annotators), i.e., an annotator could only see the Czech sentence and Czech class members of the classes offered, and similarly for English annotation items (Fig. 1).

The annotation task was to determine the word sense of the marked verb occurrence in the text. More precisely, the annotators have been asked to check one or more of the verb classes offered, or to check the "Other" option, which has been always present. They were not allowed to check "Other" *and at the same time* one or more of the classes for any of the annotation items (i.e., sentences with a particular verb occurrence) being annotated.

The five annotators were given 449 English verbs (139 different) used in 358 different sentences and another five annotators were given 448 Czech verbs (187 different) used in 358 translated sentences.[2] The annotators have not been given the definition of synonymy as presented in CzEngClass. Instead, they have been instructed to determine the correct class intuitively by extracting the "meaning" of the presented tentative class from all the members (although they can be ambiguous just by seeing the lemma, or base form as recorded in the class), much in the way the "meaning" of a WordNet synset can be understood from all the synset members.

On the other hand, several classes may actually have the same meaning, although a few verbs might only be in one or the other, with a majority of verbs being shared between the classes. This happens due to the fact that the CzEngClass lexicon is still under development, as described in (Urešová et al., 2018a), and two classes might still be merged later (we will discuss the side effect of the present annotation experiment in determining which classes should probably be merged in Sect. 6). For these reasons, annotators were allowed to assign more than one class to the verb occurrence in text. Conversely, they could use "Other" when no other class seemed to adequately fit the meaning of the verb in question.

The annotators marked the class(es) they believed to fit the meaning of the verb in question (marked by the double square brackets), or selected the "Other" class by simply putting and 'x' after the colon delimiting the class ID (or the "Other" label).

Since each annotation item has been annotated by three annotators, we have automatically determined the final annotation by taking a majority (of the three) as the final assignment of the verb to its class(es). This majority voting has been done per class offered at each annotation item and language separately.

After this step, each annotation pair has been marked for a *complete match* (when all classes marked agreed between the two languages, or "Other" was selected on both sides) or for a

[2] Some sentences contained two or more verbs, sometimes linked to the same verb in the paired sentence, thus the difference in the number of annotation items. In fact, there were 450 aligned annotation items altogether.

```
ID: n01020004#7
Lemma:  see
Sentence:  Previously the jets had only been [[seen]] by bloggers.
vec00032:  assume, believe, consider, feel, figure, see, think
vec00092:  anticipate, assume, believe, call, envisage, expect, figure,
foresee, predict, presume, project, see, suppose
vec00115:  examine, eye, follow, follow_up, look, monitor, observe, pursue,
see, study, trace, track, track_down, watch
vec00192:  analyze, determine, discover, find, find_out, learn, see
other:
```

Figure 1: Example annotation item as presented to the annotators

partial match (when at least one of the classes was the same for both languages). Any complete match was also considered a partial match, but indeed not vice versa.

4.2 Interannotator Agreement

Having three annotation per annotated item, we computed both Cohen's kappa (pairwise) and Fleiss' kappa (for all annotations).

Evaluation has been done not on the class level, but on the individual membership question annotation level. In other words, the evaluated *annotation decisions* have been just yes/no for each class offered (or the "Other" label). For each annotation item, there have been at least three such decisions. The example in Fig. 1 shows five annotation decisions, four for the classes `vec00032, vec00092, vec00115, vec00192` or one for `other`. We have separately evaluated three types of agreement: on all decisions points (1762 yes/no decisions in Czech, 1763 in English), on decisions for classes offered to the annotators (i.e., excluding the `other` choices; 1314 decisions) and for an estimate on non-news data, we limited the evaluation to classes excluding "verbs of saying" (which we noticed there have been many in the PUD corpus used, and these are apparently easy to do, padding the numbers for the better too much; excluding further 400 decisions points for those, this evaluation has been performed for 914 annotation decisions).

4.2.1 Interannotator agreement: Cohen's kappa

Cohen's kappa, used for comparing annotations pairwise, is defined as follows:

$$\kappa = (p_0 - p_e)/(1 - p_e) \tag{1}$$

where p_0 is the observed agreement, and p_e the expected agreement based on the two annotations. Results are summarized in Table 2.

From these figures, it is clear that agreement on the individual annotation decisions is low, especially for English the group of annotators labeled A_3 vs. the other two annotator groups. Further inspection has shown that in many cases, the decisions have not been "off" completely, but using only one sentence context (and intuition only based on the composition of the class, and without access to the often defining words in the other language) is difficult. The way the PUD sentences and the classes offered have been selected also contributed to the difficulty (only ambiguous classes have been used). Nevertheless, for the purpose of the syntax correlation

Czech Annotator group	A_2	A_3	A_{random}
A_1	0.630 (0.624, 0.587)	0.650 (0.663, 0.603)	0.034
A_2		0.641 (0.660, 0.572)	0.056
A_3			0.032

English Annotator group	A_2	A_3	A_{random}
A_1	0.600 (0.564, 0.605)	0.514 (0.527, 0.480)	-0.007
A_2		0.461 (0.452, 0.427)	-0.006
A_3			0.053

Table 2: Interannotator agreement: Cohen's kappa, pairwise; the three figures for each pair of annotators correspond to all (non-other, non-other and non-speech) decision points. The last column shows Cohen's kappa computed against a random baseline, for each annotator group (for all annotation items), run as a sanity check.

experiment, this seemed sufficient due to the majority voting mechanism as described in Sect. 4; the comparison against the random baseline also shows that the annotators do contribute significantly their langauge and world knowledge.

4.2.2 Interannotator agreement: Fleiss' kappa

Fleiss' kappa can be used in a multiannotator setup with n (three in our case) annotators and N datapoints (1762/1763, 1314 and 914 datapoints for the three evaluations, as defined in Sect. 4.2.1):

$$\kappa = (\bar{P} - \bar{P_e})/(1 - \bar{P_e}) \tag{2}$$

where

$$\bar{P} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n(n-1)}(n_{iYes}(n_{iYes}-1) + n_{iNo}(n_{iNo}-1)), \tag{3}$$

$\bar{P_e} = {p_Y}^2 + {p_N}^2$, $p_Y = \frac{1}{Nn}\sum_{i=1}^{N} n_{iYes}$ and $p_N = \frac{1}{Nn}\sum_{i=1}^{N} n_{iNo}$. The n_{iYes} (n_{iNo}) counts are defined as the number of times the i-th annotation decision has been annotated Y and N, respectively (i.e., sum of n_{iYes} and n_{iNo} is n). The Ys correspond to an 'x' being marked by an annotator next to the class they deemed as being the one corresponding to the meaning of the verb in question in the annotated sentence (Fig. 1), and an 'N' means the class (incl. the `other` one) has *not* been marked by the 'x'.

Table 3 shows the values of Fleiss' kappa for the three agreement evaluations performed.

The Fleiss' kappa figures show that the annotator agreement is lower for English, which is explained by the annotator group A_3 disagreement with A_1 and A_2. They also show that contrary to expectations, the exclusion of the `other` label did not lower the agreement dramatically; for

Czech

# of decisions	p_Y	p_N	$\bar{P}$	$\bar{P}_e$	κ
1762	0.353	0.647	0.836	0.543	**0.640**
1314	0.389	0.611	0.833	0.524	**0.649**
914	0.332	0.668	0.817	0.556	**0.587**

English

# of decisions	p_Y	p_N	$\bar{P}$	$\bar{P}_e$	κ
1763	0.352	0.648	0.783	0.544	**0.524**
1314	0.362	0.638	0.774	0.538	**0.511**
914	0.308	0.692	0.788	0.574	**0.502**

Table 3: Interannotator agreement: Fleiss' kappa; the three figures for each pair of annotators correspond to all (non-other, non-other and non-speech) decision points

English it did but for Czech it increases, even if insignificantly. The additional exclusion the frequent "verbs of saying" in direct and indirect speech constructions did lower the kappa, but only marginally and more in Czech than in English.

The effect of the A_3 group has been minimized when preparing the final dataset for the experiments with syntax correlation by simply using majority voting (the two or three of the 3 decisions that agreed ('Y' vs. 'N', or 'x' vs. nothing) have been selected and used as "gold data".

5 Correlation of Syntax and the Synonym Class Annotation

5.1 The Data

The data have been prepared as described in Sect. 4. For each annotated item (a verb in a sentence context), for both languages (Czech and English), we knew the "gold" class(es) annotated, and therefore also knew if the Czech and English class(es) are the same or not. Due to the possibility of having more than one class per verb, we have established two variants of class agreement: complete match (all classes selected for the Czech verb match all the classes selected for the aligned English verb), and partial match (at least one class matches across the languages). We have computed the correlation to syntactic properties, as described below, always for both.

The syntactic features used to check if there is any correlation between them and the agreement (or disagreement) between the two languages have been the presence or absence of the verb's "core arguments" as defined in the UD specification (Zeman, 2017).[3] The following dependency relations (DEPRELs), as used in the Czech and English UD annotation, are considered core arguments: `nsubj, csubj, obj, iobj, ccomp, xcomp, expl`, including cases when they have a language specific extension (e.g., `nsubj:pass or obj:caus`). In addition, `obl:arg and obl:agent` are also considered core arguments.[4]

[3] `http://universaldependencies.org`

[4] For detailed UD syntactic relations see http://universaldependencies.org/u/dep/, where `nsubj` is a nominal which is the syntactic subject, `csubj` is a clausal syntactic subject, `obj` is an object, `iobj` is an indirect object (mostly in the dative case), `ccomp` (clausal complement) is a dependent clause which functions like an object of the verb, `xcomp` (open clausal complement) is a predicative or clausal complement without its own subject, `expl` (expletive) captures expletive or pleonastic nominals that appear in an argument position of a predicate but which do not themselves satisfy any of the semantic roles of the predicate, `obl:arg` is used for prepositional objects, and `obl:agent` for agents of passive verbs.

Example of the resulting data (four annotation items) is in Table 4.

			match	
item	cs core args – en core args	cs ID – en ID	complete	partial
...	...	...	...	...
139	ccomp nsubj – nsubj	n01145028#11 – n01145028#15	1	1
140	ccomp nsubj – nsubj obj	n01059054#19 – n01059054#24	0	0
141	ccomp nsubj – nsubj xcomp	n01108003#2 – n01108003#3	0	1
142	ccomp nsubj – nsubj xcomp	n01136006#4 – n01136006#4	0	0
...	...	...	...	...

Table 4: Section of the correlation dataset (ordered by core arguments)

Complete match always implies partial match, as in item 139. Items 140 and 142 did not exhibit even partial match between the class assigned to the Czech and English verb.

Such data can be used in several ways. We have looked at them from two points of view, to get some insight into the following two research questions:

- if we know the core argument pattern, or perhaps only some of its features, can we predict whether there will be an agreement in assigning the Czech and English class to a given occurrence of a verb (even if annotated independently of each other, as described in Sect. 4)?
- if we know the core argument pattern, or perhaps only some of its features, can we predict the difficulty of predicting a match?

While answering the first question might give us the possibility to extract a set of argument patterns and/or their features, for which we can get better predictions and thus need less human annotation (because we can then use just a single class for both Czech and English annotation in a parallel corpus, achieving higher accuracy upfront), answering the second question in the positive would mean that we can extract a set of core argument patterns that are difficult to predict even from the point of view of agreement (regardless whether there is one or not), and those will definitely need human intervention and investigation.

5.2 Prediction of a Match

For the investigation of the first question, we use the simple conditional probability of a match between the Czech and English annotation given the core argument structure (or some projection into simpler features).

We have investigated systematically several such distributions, conditioned on the following:

- the whole pattern
- one of the arguments, or some of its features (passivization)
- a combination of two or more arguments, either on Czech side, the English side, or both

Some of the above conditionings split the data into many parts, some of them very small (with a long tail of singletons, as usual). We have thus excluded any split that displayed less than 5 datapoints.

core argument(s) configuration	configuration probability	probability of a complete match
exact cs–en: `ccomp - nsubj`	0.018	0.875
cs side contains: `csubj`	0.016	0.860
en side contains: `iobj`	0.016	0.860
exact cs–en: `- nsubj obj`	0.013	0.833
exact cs–en: `- nsubj`	0.018	0.750
>2 arguments on both sides	0.024	0.727

core argument(s) configuration	configuration probability of	probability a partial match
exact cs–en: `nsubj obj iobj - nsubj obj xcomp`	0.011	1.000
cs side contains: `csubj`	0.016	1.000
en side contains: `iobj`	0.016	1.000
exact cs–en: `ccomp - nsubj`	0.018	1.000
exact cs–en: `nsubj ccomp - nsubj`	0.051	1.000
exact cs–en: `nsubj ccomp - nsubj ccomp`	0.076	1.000
cs side contains: `nsubj ccomp`	0.156	0.971
en side contains: `nsubj ccomp`	0.136	0.934
cs side contains: `ccomp`	0.249	0.911
>2 arguments on both sides	0.024	0.909
en side contains: `ccomp`	0.153	0.899
>2 arguments on cs side:	0.084	0.895
exact cs–en: `nsubj obj -`	0.016	0.857
2 arguments on both sides	0.251	0.832

Table 5: Best predictors of a match in terms of core arguments and/or core argument features (complete match in the upper part, partial match in the lower part)

Core argument combinations with high probability (over 70%) of a match between Czech and English synonym class when annotating a given verb in a textual context are listed in Table 5.

It is clear from this table that while some of the core argument features predict a match quite well, they are very infrequent (and thus do not help much overall) due their specificity, e.g., "no argument configuration" on the Czech side (for `- nsubj obj` and `- nsubj`), or the rare `csubj` on the Czech side, etc. Only when we resort to partial matches (hoping that after the synonym dictionary is cleaned and merger candidates actually merged, they will become complete matches), there are some more frequent situations which result in high probability of a partial match, e.g., if `ccomp` appears on the Czech side (which happens in 24.9% of the annotated sentences), or the situation when there are exactly 2 arguments on both sides (which happens in 25.1% of them). Closer inspection has shown, however, that most of the cases where core arguments contain `nsubj` and/or `ccomp` on either the Czech or English side are, not surprisingly, the "verbs of saying", which are easy to classify.

None of the other combinations of core argument configurations and/or their features (such as passives) showed more than 70% prediction of a complete match (80% in the case of a partial match), excluding those occurring less than five times, as stated above.

5.3 Correlation (Mutual Information)

In the previous section, we have used the conditional probability of a match given a core argument configuration as a measure to find "good" predictors. This measure is intuitive, but it does not find *globally* "good" predictors. This is better accomplished by using correlation measures. We have used both the Pearson correlation and Mutual Information (Cover and Thomas, 2006). Since the results are very similar, we will resort to Mutual Information only to determine which core argument configurations predict the (complete or partial) match better.

Mutual Information (MI) is defined as the entropy reduction when a variable X is used to predict Y (as opposed to not using the variable X). It can thus be used to compare the contribution of various variables X_k to the prediction of Y. Using probability distribution(s), it is defined as

$$MI(X_k, Y) = \sum_{i=1}^{|X_k|} \sum_{j=1}^{|Y|} p(x_{k_i}, y_i) log_2 \frac{p(x_{k_i}, y_i)}{p(x_{k_i})p(y_i)} \quad (4)$$

MI is a "global" measure, thus it helps to determine which X_k works best over all data, since it includes the proper distribution of weights, equal to the joint probability of the various configurations ($= X_k$s) and the prediction (0 or 1, in our case). MI has always a positive value. In our case, the higher it is, the more the tested core argument configuration of features as represented by the X_k variable helps to predict the match.

We have computed Mutual Information between the same large set of possible core argument configurations as in Sect. 5.2, using the presence or absence of the given configuration or feature (represented as 0 or 1) as the predictor, and also computed the Mutual Information of the set of full individual core argument configurations. The results are in Table 6.

Using all possible configurations reduces the prediction entropy the most, as expected. There are 128 different cs – en configurations in the data. Unfortunately, only 20 of them occur more than in 1% of the data, and the easily predicted (in)direct speech verbs occur in more than 30%, coinciding with the more frequent core argument configuration (typically containing or solely consisting of `nsubj` and `ccomp`). So while these predict the "training data" well, problems will arise on previously unseen data. This is normally alleviated by using the more robust binary features that only look at the presence of absence of individual core arguments on one or the other side (cs or en). However, as Table 6 shows, these have very low values of MI, meaning that they only marginally contribute to the prediction certainty. In addition, they mostly apply again to the (in)direct speech verbs, with only a few exceptions, such as the presence of the `obl:agent` relation in Czech or of an expletive relation (Bouma et al., 2018) on the English side.

To summarize, we consider these to be negative results, essentially confirming that syntax alone (as represented by the UD core arguments) cannot be used to tell the easy from the difficult cases of assigning a common class to an aligned pair of verbs in English-Czech translation.

6 Side Effect: Classes to be Merged

Since the CzEngClass version 0.2 lexicon used (Urešová et al., 2018b) is still under development and the classes are being created independently of each other, based on independently determined seed words (Urešová et al., 2018a), some pairs of classes are very similar.

core argument(s) configuration	MI between configuration and a complete match
all configurations	0.336
cs side contains: `nsubj ccomp`	0.010
cs side contains: `obl:agent`	0.008
cs does not contain passive, en side does (`:pass`)	0.006
exact cs–en: `nsubj ccomp - nsubj ccomp`	0.005
en side contains passive (`:pass`)	0.005

core argument(s) configuration	MI between configuration and a partial match
all configurations	0.285
cs side contains: `nsubj ccomp`	0.043
cs side contains: `ccomp`	0.032
exact cs–en: `nsubj ccomp - nsubj ccomp`	0.030
cs side contains: `nsubj ccomp`	0.022
en side contains: `ccomp`	0.015
en side contains: `expl`	0.010
cs side contains: `iobj`	0.007
>2 arguments on cs side	0.007
2 arguments on both sides	0.006
cs side contains: `csubj`	0.006
cs side contains: `iobj`	0.006
en side contains: `nsubj`	0.005

Table 6: Best global predictors (based on MI) of a match in terms of core arguments and/or core argument features (complete match in the upper part, partial match in the lower part)

The annotation as set up for the purpose of this paper (Sect. 4) can be used to identify those classes that are hard to distinguish, thanks to the fact that annotators may check multiple classes for each annotated item.

Statistics about the multiple selections suggest to merge several pairs of classes. These include the verbs introducing direct or indirect speech, which are frequent in our corpus (and which we have mentioned a few times already) and currently in two classes (they include verbs like *say, announce, declare, disclose, report, post, advise,* etc.). Other pairs (and one triple) with high annotator agreement include (only English verbs from these classes shown):

- allow, enable, permit, ... vs. approve, acknowledge, reaffirm, ...
- anticipate, forecast, foresee, ... vs. expect, await, pend, ...
- avoid,[5] bypass, circumvent, get around, ... vs. avoid, discourage, preclude, prevent
- (triple) encourage, galvanize, inspire, ... vs. spark, launch, set off, ... vs. aid, bolster, encourage, facilitate, ...

[5] An interesting example is the "avoid" example, which could go unnoticed if only the lexicon classes themselves are investigated, since it looks at the same event from different perspectives (with distinct different syntactic realization). Annotation on the PUD corpus has however revealed its closeness if not identity from the semantic point of view.

These results confirm that synonym classes might be impossible to define categorically, or mathematically speaking, as equivalence classes; it points rather to something like "soft class membership" function or other means to relate related words which we call, perhaps incorrectly, "synonyms", and it suggests to depart from thinking of every two words as strictly either being synonyms or not. This naturally leads to considerations of "weighted" synonymy, or to introduce a distance between every two (senses of) a verb (or any other word) which reflects their synonym(it)y; there are plenty of measures to experiment with, starting with cosine similarity to modified "soft" Brown classes (Brown et al., 1992) to distance measures over various types of embeddings, such as (Mikolov et al., 2013); this is however outside of the scope of this paper and will be our future work on this topic.

7 Conclusions and Outlook

We have performed an experiment using manual annotation followed by sort of "data mining" to determine whether the presence or absence of (syntactic) core arguments (as defined in the UD specification) can be used in determining the difficult cases of assignment of a synonym class (defined in semantic terms). While expecting that syntax alone cannot be used for such a prediction, we were surprised by a correlation so low.[6]

As a positive side effect outcome of the annotation experiment performed, we have clearly identified candidate class pairs (and one triple) for a merger. It remains to be seen if the small dataset that we have created could be of any use when developing automatic methods for such mergers. We believe that it can at least serve as a test corpus. This part of the experiment has also shown that it might be better (or even necessary) to introduce "soft" (or "fuzzy") membership in classes.

It has been confirmed that using bilingual data (parallel corpus) provides important information; in our experimental setting we could not automatically identify "matches" (to be used as "gold truth" for the syntax correlation experiments) without using them.

The main conclusion is that for the actual assignment of classes (i.e., not only the prediction of the *difficulty* of such assignment, as demonstrated in this paper), which is certainly a more difficult task, it will not be sufficient to look at (dependency) syntax only, and that semantically-based methods will have to be used, such as the use of automatically derived classes and their bilingual pruning, or the use of embeddings trained on large corpora.

Acknowledgments

This work has been supported by the grant No. GA17-07313S of the Grant Agency of the Czech Republic. The data used in this work have been created and are maintained in the LINDAT/CLARIN digital repository (`http://lindat.cz`), supported by the Ministry of Education, Youth and Sports of the Czech Republic (MEYS) as projects No. LM2015071 and CZ.02.1.01/0.0/0.0/16_013/0001781. In addition, some of the data used has been the result of the "T4ME" project funded by the EC as project No. ICT-2009-4-249119 (MEYS No. 7E11040).

[6] The only mild exception are "verbs of saying" (*say, announce, add, report, ...*), relatively frequent in the news texts that the PUD corpora represent, since they are easily recognizable by their syntactic structure (usually `nsubj` and `ccomp` only).

References

Alishahi, A. and Stevenson, S. (2010). A computational model of learning semantic roles from child-directed language. *Language and Cognitive Processes*, 25(1):50–93.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bouma, G., Hajič, J., Nivre, J., Solberg, P., and Ovrelid, L. (2018). Expletives in universal dependency treebanks. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 18–26, Bruxelles, Belgium. Association for Computational Linguistics.

Bowerman, M. (2002). Mapping thematic roles onto syntactic functions: Are children helped by innate linking rules? In *Mouton Classics: From Syntax to Cognition, from Phonology to Text, Vol. 2. (Original in Linguistics, 1990, 28, 1253-1289.)*. Mouton de Gruyter.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479.

Čech, R., Mačutek, J., and Koščová, M. (2015). On the relation between verb full valency and synonymy. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 68–73. Uppsala University, Uppsala, Sweden.

Cinková, S. (2006). From PropBank to EngVallex: Adapting the PropBank-Lexicon to the Valency Theory of the Functional Generative Description. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 2170–2175, Genova, Italy. ELRA.

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory, 2nd Edition*. Wiley.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA. 423 pp.

Fillmore, C. J., Johnson, C. R., and L.Petruck, M. R. (2003). Background to FrameNet: FrameNet and Frame Semantics. *International Journal of Lexicography*, 16(3):235–250.

Hajič, J., Panevová, J., Urešová, Z., Bémová, A., Kolářová, V., and Pajas, P. (2003). PDT-VALLEX: Creating a Large-coverage Valency Lexicon for Treebank Annotation. In Nivre, Joakim//Hinrichs, E., editor, *Proceedings of The Second Workshop on Treebanks and Linguistic Theories*, volume 9 of *Mathematical Modeling in Physics, Engineering and Cognitive Sciences*, pages 57—68, Vaxjo, Sweden. Vaxjo University Press.

Hartshorne, J. K., O'Donnell, T. J., Sudo, Y., Uruwashi, M., and Snedeker, J. (2010). Linking meaning to language: Linguistic universals and variation. In Ohlsson, S. and Catrambone, R., editors, *Proceedings of the 32nd Annual Meeting of the Cognitive Science Society*, pages 1186–1191, Austin, TX. Cognitive Science Society, Department of Linguistics and Scandinavian Studies, University of Oslo.

Kettnerová, V., Lopatková, M., and Bejček, E. (2012). Mapping semantic information from FrameNet onto VALLEX. *The Prague Bulletin of Mathematical Linguistics*, 97:23–41.

Kingsbury, P. and Palmer, M. (2002). From Treebank to PropBank. In *Proceedings of the LREC*, Canary Islands, Spain.

Levin, B. (1993). *English Verb Classes and Alternations: A Preliminary Investigation*. The University of Chicago Press, Chicago and London.

Levin, B. and Hovav, M. R. (2005). *Argument realization*. Cambridge Univ. Press, Cambridge.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Nivre, J., Abrams, M., Agić, Ž., Ahrenberg, L., Antonsen, L., Aranzabe, M. J., Arutie, G., Asahara, M., Ateyah, L., Attia, M., Atutxa, A., Augustinus, L., Badmaeva, E., Ballesteros, M., Banerjee, E., Bank, S., Barbu Mititelu, V., Bauer, J., Bellato, S., Bengoetxea, K., Bhat, R. A., Biagetti, E., Bick, E., Blokland, R., Bobicev, V., Börstell, C., Bosco, C., Bouma, G., Bowman, S., Boyd, A., Burchardt, A., Candito, M., Caron, B., Caron, G., Cebiroğlu Eryiğit, G., Celano, G. G. A., Cetin, S., Chalub, F., Choi, J., Cho, Y., Chun, J., Cinková, S., Collomb, A., Çöltekin, Ç., Connor, M., Courtin, M., Davidson, E., de Marneffe, M.-C., de Paiva, V., Diaz de Ilarraza, A., Dickerson, C., Dirix, P., Dobrovoljc, K., Dozat, T., Droganova, K., Dwivedi, P., Eli, M., Elkahky, A., Ephrem, B., Erjavec, T., Etienne, A., Farkas, R., Fernandez Alcalde, H., Foster, J., Freitas, C., Gajdošová, K., Galbraith, D., Garcia, M., Gärdenfors, M., Gerdes, K., Ginter, F., Goenaga, I., Gojenola, K., Gökırmak, M., Goldberg, Y., Gómez Guinovart, X., Gonzáles Saavedra, B., Grioni, M., Grūzītis, N., Guillaume, B., Guillot-Barbance, C., Habash, N., Hajič, J., Hajič jr., J., Hà Mỹ, L., Han, N.-R., Harris, K., Haug, D., Hladká, B., Hlaváčová, J., Hociung, F., Hohle, P., Hwang, J., Ion, R., Irimia, E., Jelínek, T., Johannsen, A., Jørgensen, F., Kaşıkara, H., Kahane, S., Kanayama, H., Kanerva, J., Kayadelen, T., Kettnerová, V., Kirchner, J., Kotsyba, N., Krek, S., Kwak, S., Laippala, V., Lambertino, L., Lando, T., Larasati, S. D., Lavrentiev, A., Lee, J., Lê Hồng, P., Lenci, A., Lertpradit, S., Leung, H., Li, C. Y., Li, J., Li, K., Lim, K., Ljubešić, N., Loginova, O., Lyashevskaya, O., Lynn, T., Macketanz, V., Makazhanov, A., Mandl, M., Manning, C., Manurung, R., Mărănduc, C., Mareček, D., Marheinecke, K., Martínez Alonso, H., Martins, A., Mašek, J., Matsumoto, Y., McDonald, R., Mendonça, G., Miekka, N., Missilä, A., Mititelu, C., Miyao, Y., Montemagni, S., More, A., Moreno Romero, L., Mori, S., Mortensen, B., Moskalevskyi, B., Muischnek, K., Murawaki, Y., Müürisep, K., Nainwani, P., Navarro Horñiacek, J. I., Nedoluzhko, A., Nešpore-Bērzkalne, G., Nguyễn Thị, L., Nguyễn Thị Minh, H., Nikolaev, V., Nitisaroj, R., Nurmi, H., Ojala, S., Olúòkun, A., Omura, M., Osenova, P., Östling, R., Øvrelid, L., Partanen, N., Pascual, E., Passarotti, M., Patejuk, A., Peng, S., Perez, C.-A., Perrier, G., Petrov, S., Piitulainen, J., Pitler, E., Plank, B., Poibeau, T., Popel, M., Pretkalniņa, L., Prévost, S., Prokopidis, P., Przepiórkowski, A., Puolakainen, T., Pyysalo, S., Rääbis, A., Rademaker, A., Ramasamy, L., Rama, T., Ramisch, C., Ravishankar, V., Real, L., Reddy, S., Rehm, G., Rießler, M., Rinaldi, L., Rituma, L., Rocha, L., Romanenko, M., Rosa, R., Rovati, D., Roșca, V., Rudina, O., Sadde, S., Saleh, S., Samardžić, T., Samson, S., Sanguinetti, M., Saulīte, B., Sawanakunanon, Y., Schneider, N., Schuster, S., Seddah, D., Seeker, W., Seraji, M., Shen, M., Shimada, A., Shohibussirri, M., Sichinava, D., Silveira, N., Simi, M., Simionescu, R., Simkó, K., Šimková, M., Simov, K., Smith, A., Soares-Bastos, I., Stella, A., Straka, M., Strnadová, J., Suhr, A., Sulubacak, U., Szántó, Z., Taji, D., Takahashi, Y., Tanaka, T., Tellier, I., Trosterud, T., Trukhina, A., Tsarfaty, R., Tyers, F., Uematsu, S., Urešová, Z., Uria, L., Uszkoreit, H., Vajjala, S., van Niekerk, D., van Noord, G., Varga, V., Vincze, V., Wallin, L., Washington, J. N., Williams, S., Wirén, M., Woldemariam, T., Wong, T.-s., Yan, C., Yavrumyan, M. M., Yu, Z., Žabokrtský, Z., Zeldes, A., Zeman, D., Zhang, M., and Zhu,

H. (2018). Universal dependencies 2.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal dependencies v1: A multilingual treebank collection. In Chair), N. C. C., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Palmer, M. (2009). SemLink: Linking PropBank, VerbNet and FrameNet. In *Proceedings of the Generative Lexicon Conference*, page 9–15.

Palmer, M., Gung, J., Bonial, C., Choi, J., Hargraves, O., Palmer, D., and Stowe, K. (2017). The pitfalls of shortcuts: Tales from the word sense tagging trenches. *To appear in: Lexical Semantics and Computational Lexicography*.

Pustejovsky, J. (1991). The generative lexicon. *Computational linguistics*, 17(4):409–441.

Urešová, Z., Fučíková, E., Hajičová, E., and Hajič, J. (2018a). Creating a Verb Synonym Lexicon Based on a Parallel Corpus. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC'18)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Urešová, Z., Fučíková, E., Hajičová, E., and Hajič, J. (2018b). CzEngClass 0.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, `http://hdl.handle.net/11234/1-2824`, Creative Commons - Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

Urešová, Z., Fučíková, E., Hajičová, E., and Hajič, J. (2018c). Synonymy in Bilingual Context: The CzEngClass Lexicon. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2456–2469.

Urešová, Z., Fučíková, E., Hajičová, E., and Hajič, J. (2018d). Tools for Building an Interlinked Synonym Lexicon Network. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC'18)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Wu, Y. (2017). The interfaces of Chinese syntax with semantics and pragmatics (Routledge Studies in Chinese Linguistics). *Journal of Linguistics*, 54(1):222–227.

Zeman, D. (2017). Core arguments in universal dependencies. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 287–296. Linköping University Electronic Press.

Quantitative word order typology with UD

Matías Guzmán Naranjo[1], *Laura Becker*[2]

(1) Heinrich Heine Universität Düsseldorf
(2) University of Erlangen-Nürnberg, Leipzig University

guzmanna@hhu.de, laura.becker@uni-leipzig.de

ABSTRACT

Cross-linguistic universals of word order correlations often based on the distinction basic VO and OV orders have received a lot of attention since the seminal work by Greenberg (1963), followed by e.g. Dryer (1991, 1989); Hawkins (1979, 1980, 1983); Lehmann (1973, 1974); Vennemann (1974, 1975). However, there have been quantitative studies (e.g. Chen and Gerdes, 2017; Dunn et al., 2011; Liu, 2010) focusing on a small number of languages (Celano, 2014), or insisting on canonical word order for every language. The aim usually is to find crosslinguistic word order correlations on the basis of this canonical order. How to determine the latter for any language is, however, highly problematic and potentially misleading for a number of languages, as was already argued convincingly in Mithun (1992): it means that stricter OV order languages such as Japanese are treated like flexible OV order languages such as German. Despite some strong crosslinguistic correlations based on canonical word order that could be confirmed in independent samples, it is still not clear whether these effects can reliably be modelled as categorical or whether we should rather treat them as gradient. This is what we propose in the present study: We explore the question of whether word order tendencies between the verb and its arguments may have some influence on the orders between nouns and their dependents, and whether these tendencies are cross-linguistic or language specific.

KEYWORDS: word order universals, word order correlations, gradient tendencies.

1 Background and motivation

Since Greenberg (1963), crosslinguistic word order correlation and related questions have received a lot of attention in language typology (Cristofaro, 2018; Dryer, 1992, 2009, 2019; Hawkins, 1994, 2014; Payne, 1992; Siewierska, 1988; Song, 2009), to name just a few. Examples of robust crosslinguistic generalizations include the following correlations between the verb-object order and the order of other elements in the clause (taken from Dryer (1991, 1992, 2009):

VO	OV
prepositions	postpositions
postnominal relative clause	prenominal genitive
prenominal article	postnominal article
verb - adverb	adverb - verb
clause-initial complementizer	clause-final complementizer

Table 1: Examples of crosslinguistic word order correlations.

Some of these correlations were argued to be based on a general preference for a head-dependent order within a given language. For instance, Hawkins (1983) argued for a so-called "cross-category harmony", meaning that we often find verb-initial languages with mostly all of the dependents following their heads, while verb-final languages should mostly have all dependents preceding their heads. SVO languages were situated in the middle and expected to feature some dependents before and some following their heads. Dryer (1992, 2009), on the other hand, shows that the explanations in terms of head-dependent orders cannot capture all the crosslinguistic patterns found. Therefore, he argues for a "branching directory theory", according to which word order correlations reflect a tendency for languages to be consistently left-branching or right-branching. Two other main types of explanations for the correlation patterns found have to be mentioned: Hawkins (1994, 2014) offers parsing-based explanations in terms of structures with shortest constituent recognition domains. Bybee (1988); Aristar (1991); Cristofaro (2018), on the other hand, show that what may be taken as a correlation between two structures on the synchronic level, are rather diachronically related structures which is why we cannot speak of correlations at all in some cases (e.g. the orders between the noun and genitives as well as with relative clauses). Nevertheless, the main idea of a basic order of different elements for the sake of consistency within a given language is taken up in Dryer (2019), who proposes 5 surface principals that account for the crosslinguistic trends, one of which being being "intracategorial harmony", i.e. the general preference of different types of nominal elements occurring on one side of the noun within a language.

What this brief overview shows is that even though the concept of basic word orders has been discussed and criticised (Mithun, 1992; Payne, 1992; Siewierska, 1988), more recent studies that seek functional explanations for word order correlations still take it for granted that we can determine the basic word order of any given language (e.g. Dunn et al., 2011; Dryer, 2019; Hawkins, 2014). To give an example, languages like German or Spanish are then assigned SOV or SVO orders (respectively), even though both languages allow for most possible orderings. This leads to a situation where a language like Japanese, which is consistently a OV language ends up in the same group as German, which despite being an OV language, allows for all possible orderings (with respect to S, V and O). Similarly, it is often the case that languages have both pre- and postpositions, but this is often not taken into account. In other words, languages

are classified categorically with respect to certain word order properties, even though we almost always find gradient variation for these features within single languages.

The aim of this study is to present a new method which could help remedy this shortcoming. The main point is that we do not need to make such categorical choices; rather, we should look at the proportions with which languages use VO vs. OV orders, and compare those proportions with the proportions of other word orders (e.g. pre- vs. postpositions). Thus, instead of saying that a language is OV, we can say that it uses OV structures in 90% of cases, while 10% of cases have a VO structure.

Even though there are a number of quantitative studies on word order variation within and across languages using dependency treebanks, most of these have slightly different objectives, e.g. examining word order freedom and zooming in on its correlation with the availability of case marking across languages (Futrell et al., 2015), or investigating the evidence for dependency length minimization for different types of head-dependent relations (Gulordava, 2018). Another study that uses dependency treebanks for crosslinguistic comparison is Liu (2010), which investigates 20 languages for the prevalence of head initial vs head final in them. We argue that this methodology can be further expanded to include more fine-grained comparisons, and establish correlations across orderings of different dependents, returning to the starting point of word order typology. Somewhat related is the study by Chen and Gerdes (2017), who classify languages according to dependency structure also using the Universal Dependencies Treebank for 20 languages. This study, however, does not explore word order universals, focusing on the distance between languages.

2 Datasets

To explore gradient word order correlations we use the Universal Dependencies Treebank (Nivre et al., 2016) version 2.2, which as of July 1st, 2018 contains 122 treebanks for 71 languages.

We are aware of the fact that this dataset has several shortcomings when used for language typology. First, there is relatively little subfamily variation. Most languages in the sample belong to an Indo-European subfamilies, and the corpora for Non-Indo-European languages are smaller (e.g. Bambara with 13K tokens) than the datasets for languages like English (586K tokens) or Russian (1247K tokens). Typological studies usually take a lot more care in selecting a balanced sample of languages (Bickel, 2008; Dryer, 1989, 2019). However, despite this clear issue, the results we obtain from looking at the Universal Dependency dataset serve as a good starting point for future work on quantitative word order correlations. As treebanks for more languages from other families and geographical locations become more readily available, one can easily expand on this study, and see whether results confirm or disprove these initial findings. The aim of this study is to present what we believe is an innovative technique, even if our results hold for the language sample of the UD treebanks and cannot be generalized as true *universal* tendencies. The second potential objection is that we entirely depend on the annotation schemes used by the creators of the UD treebanks, which is not yet perfectly consistent. However, as the UD project aims at having an annotation scheme which is applicable to different languages and comparable across them, the UD treebanks certainly offer a robust crosslinguistically comparable annotation.

3 Methodology

The UD project offers treebanks for 70 languages of 20 sub families, 8 of which are Indo European. We first combined the available treebanks for all languages. The families and the number of languages in each subfamily were the following: Afro-Asiatic (4), Altaic (6), Armenian

(1), Austronesian (2), Baltic (2), Basque (1), Celtic (2), Creole (1), Defoid (1), Dravidian (2), Germanic (9), Greek (2), Indo-Iranian (6), Pama-Nyungan (1), Romance (9), Swedish Sign Language (1), Sinitic (2), Slavic (12), Uralic (5), Viet-Muong (1).

We extracted the dependents from the treebanks for each noun, for each verb, and whether these dependents preceded or followed their heads. We only considered verb dependents with one of the following part-of-speech tags: NOUN, VERB, PROPN (proper noun), PRON (pronoun) and AUX (auxiliary). We considered all noun dependents. We made this decision in order to restrict correlations to content words, which seemed more likely to occur crosslinguistically. This gives us a count for each language: for instance, of how many times the determiner follows and precedes a noun, or of how often objects follow or precede the verb, etc. From these absolute occurrences of different types of head-preceding and head-following dependents, we calculated the proportion of a given dependent following its head (noun or verb).

We took into account the following types of verb dependents:

- *advcl*: adverbial clause modifiers
- *advmod*: adverbial modifiers (non clausal)
- *nsubj*: nominal subject (noun phrase which acts as subject of the verb), first core argument of the clause
- *obj*: (direct) object of a verb, second core argument of the clause
- *obl*: oblique, or non-core argument of the verb

The noun dependents we considered are the following:

- *advcl*[1]: adverbial clause modifiers
- *acl*: clausal modifiers of nouns
- *amod*: adjectival modifiers
- *case*: used for any case-marking element which is treated as a separate syntactic word (mostly prepositions, but also postpositions, and clitic case markers)
- *compound*: relation used to mark noun compounding
- *det*: nominal determiners
- *nmod*: nominal modifiers of other nouns (not appositional)
- *nummod*: numeral modifiers of nouns

Clearly, the use of these dependency relations has some benefits as well as potential issues. An advantage is that the UD treebanks inherently aim at defining these relations in such a way that they are crosslinguistically applicable. However, it is not the case that, at this point in the development of the UD treebanks, all treebanks use these relations consistently, and since checking this is prohibitively complicated and time consuming, we have to assume that the relations used and the annotation schemes are comparable to some extent.

[1] We sometimes mark this as *n_advcl* to distinguish it from the adverbial clause modifiers for verbs.

4 Results

We explore several questions in this section. First, we examine the distribution of proportions for each of the dependents that are included in this study. The distribution of proportions is a first sanity check, to make sure our data aligns with what we know about these categories from previous studies. Second, we examine the intracategorial harmony (Dryer, 2009, 2019) of orderings of the noun dependents and verb dependents. The final question is whether noun dependent ordering proportions are predictable from verb dependent ordering proportions, and vice versa. Although related to the crosscategorial harmony proposed in Hawkins (1983), it is a novel question which directly extends the classical implicational universals that have been established in word order typology. Since we examine proportions of head-dependent orders, single languages correspond to single data points. Therefore, we address the languages collectively and not separately in this section in order to assess crosslinguistic tendencies of word order correlations between different types of heads and dependents. For brevity, we some times refer to *dependent ordering proportions* simply as *noun dependent* or *verb dependent*.

4.1 Distributions

We first explore the distribution of all dependents and their position with respect to their heads. Figures 1 and 2 present the distribution of verb and noun dependents for all languages, with the highest proportions of preceding dependents on the left and the highest proportions of following dependents on the right. We can observe some clear global trends. First, for verb dependents, there is a pronounced preference for subjects to be preverbal. This is likely due to the fact that subjects are often topics and thus given information, which has been shown to generally precede new information in the sentence (Gundel, 1988; Lambrecht, 1994; Arnold et al., 2000; Taboada and Wiesemann, 2010; Junge et al., 2015). For both direct and oblique objects, on the other hand, we see a bimodal distribution with a preference for both categories following the verb, and obliques being somewhat more flexible. Adverbial clauses show a similar preference for postverbal position, but less pronounced than objects and obliques. Finally, adverbial modifiers are predominantly preverbal.

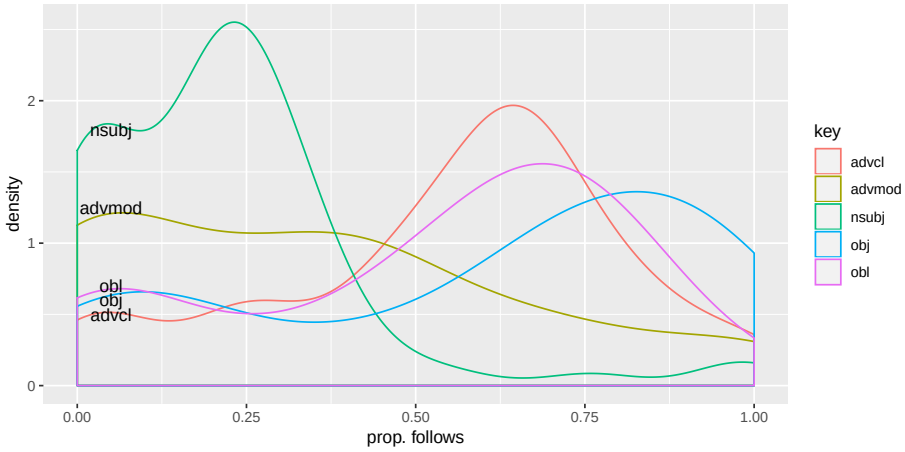


Figure 1: Density distribution for verb dependents.

For noun dependents, Figure 2 shows that the situation is somewhat different. The top plot

illustrates the proportions of prenominal (left) and postnominal (right) clausal, adjectival modifiers, nominal, numeral, and adverbial clause modifiers of nouns. All these dependents have a preference for being either post- or preverbal, but they also appear in the other position, respectively. In the bottom plot, on the other hand, we see two other types of distributions: case marking words (i.e. mainly adpositions) and compounds have no clear preference for either position; while determiners show a very strong preference within single languages as well as across languages to precede the head with only few exceptions. Thus, there seems to be a clear difference between these two distribution types of dependents: those with and those without strong preferences for prenominal or postnominal occurrence.

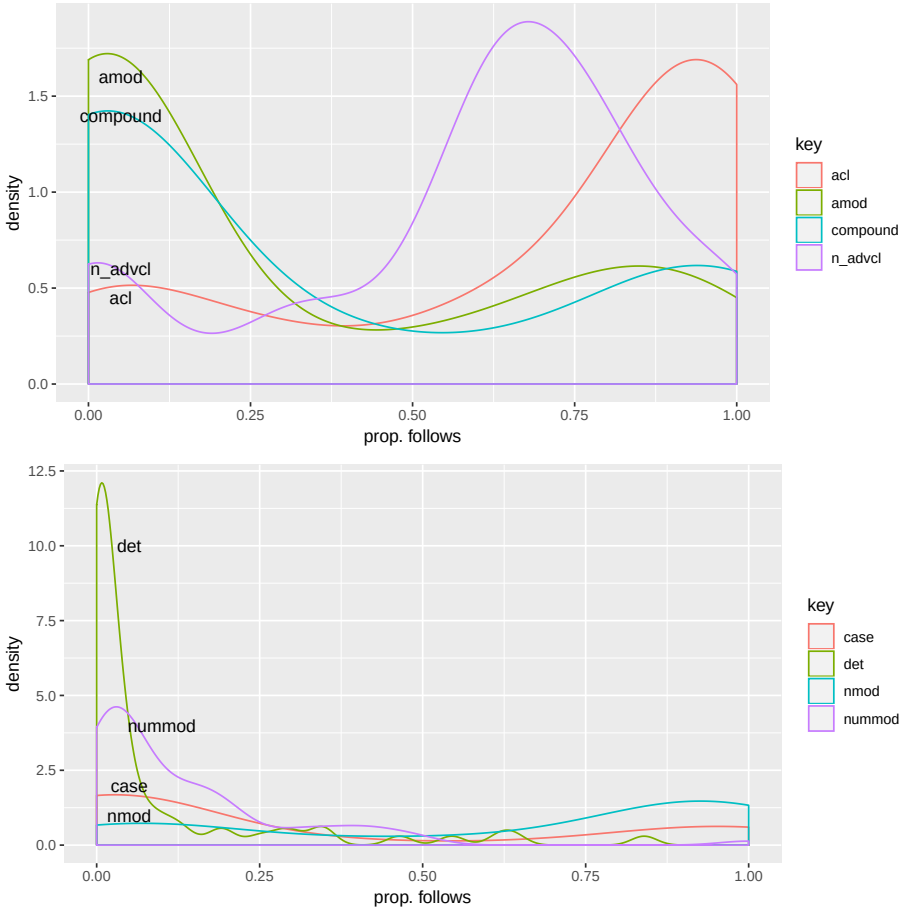
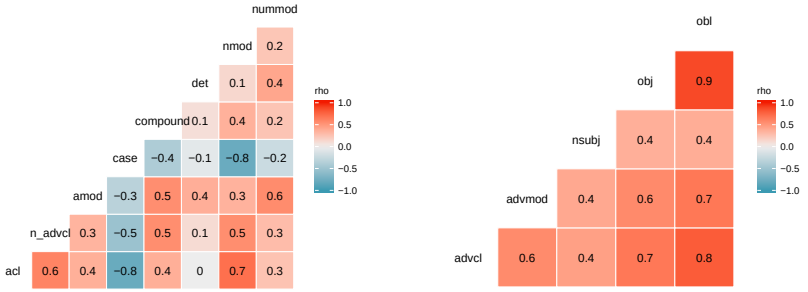


Figure 2: Density distribution for noun dependents.

4.2 Basic correlations

First, we examine the intracategorial correlations of head-dependent orders. Figure 3 presents the correlations between dependents of the verb as well as of the noun for all languages in our dataset. For nouns (a), we observe a strong positive correlation (red) between the position of the nominal modifier (*nmod*) and clausal modifiers (*acl*). This means that both indeed tend



(a) Correlations between noun dependents. (b) Correlations between verb dependents.

Figure 3: Correlations between dependents.

to occur on the same side of the noun. We also see negative correlations (blue) between case marking elements (*case*) and both clausal and nominal modifiers, meaning these tend to occur at the opposite side of the noun. What this means is that we often find structures in which the *case* element connects the head noun and the dependent noun (*nmod*, as in "the house of the major", in which the case element ("of") precedes its head ("major"), which is at the same time the nominal modifier of the head noun "house" and follows the latter. On the other hand, structures with e.g. a following *nmod* and *case* element (like "the house the major of") are infrequent in our dataset. This observation relates to Himmelmann (1997, 159-188), who discusses a class of nominal linking elements (called "linking articles") that are used in a number of languages to indicate nominal modification by other nouns, adjectives, and clauses. Interestingly, this linking element always occurs between the two elements, and not at the edge of the noun phrase.

For verb dependents, we see that there are no negative correlations, but at least two strong positive correlations, namely between direct objects (*obj*) and oblique objects (*obl*), as well as obliques and adverbial clause modifiers. That the different types of objects and clausal modifiers tend to occur on the same side of the noun supports that there is a general intracategorial harmony. On the other hand, the order of subjects being less strongly correlated to other verb dependents again points towards a stronger information structural effect motivating the position of subjects.

The next step is to consider intercategorial correlations between noun and verb dependents, as is shown in Figure 4. We find a strong correlation (red) between the position of oblique objects (*obl*) with respect to the verb, and the direction of adverbial clauses modifying a noun (*n_advcl*), clausal modifiers of the noun (*acl*), and nominal modifiers (*nmod*). Similarly, there is a strong negative correlation (white) between obliques and the position of case markers. The position of the object with respect to the verb (*obj*) also correlates with the position of clausal modifiers and nominal modifiers.

Perhaps somewhat interesting is that we do not see any strong correlations between the position of the subject with respect to the verb and other head-dependent features. This is not completely surprising in the light of Vennemann (1974, 1975); Lehmann (1973, 1974); Dryer (1991), but it is a nice corroboration that subject position has more to do with information structure, than

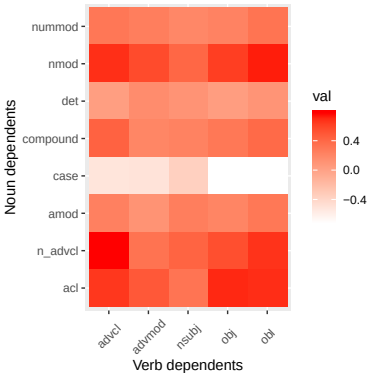


Figure 4: Correlations between noun and verb dependents.

other things. We confirm this observation in the models in the next section.

4.3 Models

In order to test for deeper correlations between these variables, we fitted beta regression models to the data, using the subfamily of the languages as a random effect. This should, at least to a certain extent, control for subfamily biases. Since in some cases the proportions of follows or precedes were equal to 1 or 0, and because beta regression does not allow for a dependent variable containing values of 1 or 0, but only values between 1 and 0, we transformed the dependent variable for every model using the technique described in (Smithson and Verkuilen, 2006). For each model, we calculated the marginal and conditional R2 values following the method developed by (Nakagawa and Schielzeth, 2013; Nakagawa et al., 2017). The marginal R2 value is the proportion of the variance explained the fixed effects alone, and the conditional R2 value is the proportion of the data explained by both the random and fixed effects. Using these two metrics, we can examine how much the fixed effects correlate with the dependent variable, and how much the variable is explained by subfamily bias.

Table 2 contains the models for noun dependents. Each row represents one model with the dependent variable in the leftmost column, then the intercept, and then the coefficients of the significant predictors. The cells marked in gray correspond to the predictors which did not reach statistical significance in the models. Strikingly, *obl* is the most frequent significant predictor, appearing in 7 out of the 10 models, even above *obj*, which proved to be significant as a predictor in only 4 of the models. This is a somewhat unexpected result, since most previous work on word order typology focused on the position of the direct object with respect to the verb, rather than the position of oblique objects with respect to the verb. One possible explanation for this fact is that in 41 out of the 70 languages, obliques are more frequent than direct objects. Another potential explanation for a difference between direct and oblique objects in predictive power could be a difference in realization as lexical noun or proform. This remains to be tested, however.

Another important result from the models in Table 2 is that there is a large difference in the predictability of the proportions of the noun dependents. The models for clausal noun modifiers (*acl*) and noun adverbial clause modifiers (*advcl*) explain a large amount of variance just with the fixed effects. These are factors which correlate with other variables in the corpus, but for

predicted	intercept	advcl	nsubj	nsbuj:obj	obj	obj^2	obj:obl	obl	obl^2	R2_m	R2_c
acl	0.02	2.02	-1.43		6.39	-3.81				0.462	0.462
n_advcl	-1.29				0.94	-5.45		3.25		0.428	0.555
amod	-1.59							1.56		0.076	0.362
case	0.5							-2.48		0.099	0.67
compound	-1.63	1.99								0.111	0.285
det	-2.88		0.74	-9.36	-0.11			2.10	3.26	0.170	0.170
nmod	-0.95	3.71			-5.31		7.20	-1.36		0.246	0.720
nummod	-2.66							1.64		0.079	0.409

Table 2: Coefficients and R2 values for models predicting noun dependents.

which there are no strong subfamily biases. In the models for adjectival modifiers (*amod*), nominal modifiers (*nmod*) and numeric modifiers (*nummod*), the fixed effects explain a small portion of the variance, while the random effects (subfamily) explains a relatively large amount of variance. These are cases where the subfamily is the main explanatory factor. Finally, in the models for case marking elements (*case*) and determiners (*det*), neither the fixed nor the random effects explain much of the variance.

Table 3 shows the models predicting verb dependents from noun dependents. Here, we see that the noun adverbial clause modifier (*n_advcl*) is the most common significant predictor (we excluded it as a predictor when *advcl* was also the dependent variable). The other important observation is that the models for both direct objects (*obj*) and oblique objects (*obl*) are very similar. They have similar coefficients, and the same two significant predictors (*advcl* and *case*), and their R2 values are also close to each other. The main difference is that for *obj*, the effect of the language subfamily seems to be larger. Subjects are the least predictable verb dependents given other verb dependents as predictors. This is most likely due to the fact that the subject position is heavily influenced by the information structure of the sentence.

predicted	intercept	acl	n_advcl	case	compound	nmod	R2_m	R2_c
advcl	-0.76				0.72	1.57	0.15	0.528
advmod	-2.07		1.65			0.97	0.240	0.240
nsubj	-1.17	-1.54	2.27	-1.26			0.161	0.320
obj	-0.30		2.86	-2.15			0.433	0.634
obl	-1.05		2.92	-1.64			0.445	0.513

Table 3: Coefficients and R2 values for models predicting verb dependents.

Finally, Figure 5 presents the three best models for verb (left column) and noun (right column) dependents: predicting *acl*, *nmod*, *advcl* (in both), *obj* and *obl*. We see the observed vs fitted values for all six models. First, the models for verb dependents are a better fit to the data. For noun dependents, we see that there are several outliers for *advcl* and *acl* from mostly Non-European languages, the overall fit still being relatively good.

5 Conclusion and Outlook

This study had two main objectives. First, and most importantly, we present a new method for investigating word order universals by making use of treebanks. Even though our results may still be somewhat influenced by the bias in the UD Treebank towards Indo-European languages, a similar approach with a more balanced, and larger corpus can provide more accurate and differentiated results than previous studies based on categorical word order distinctions. This objection not withstanding, we want to emphasize that our results generally agree with previous observations in the literature. This fact leads us to be relatively confident that our results should

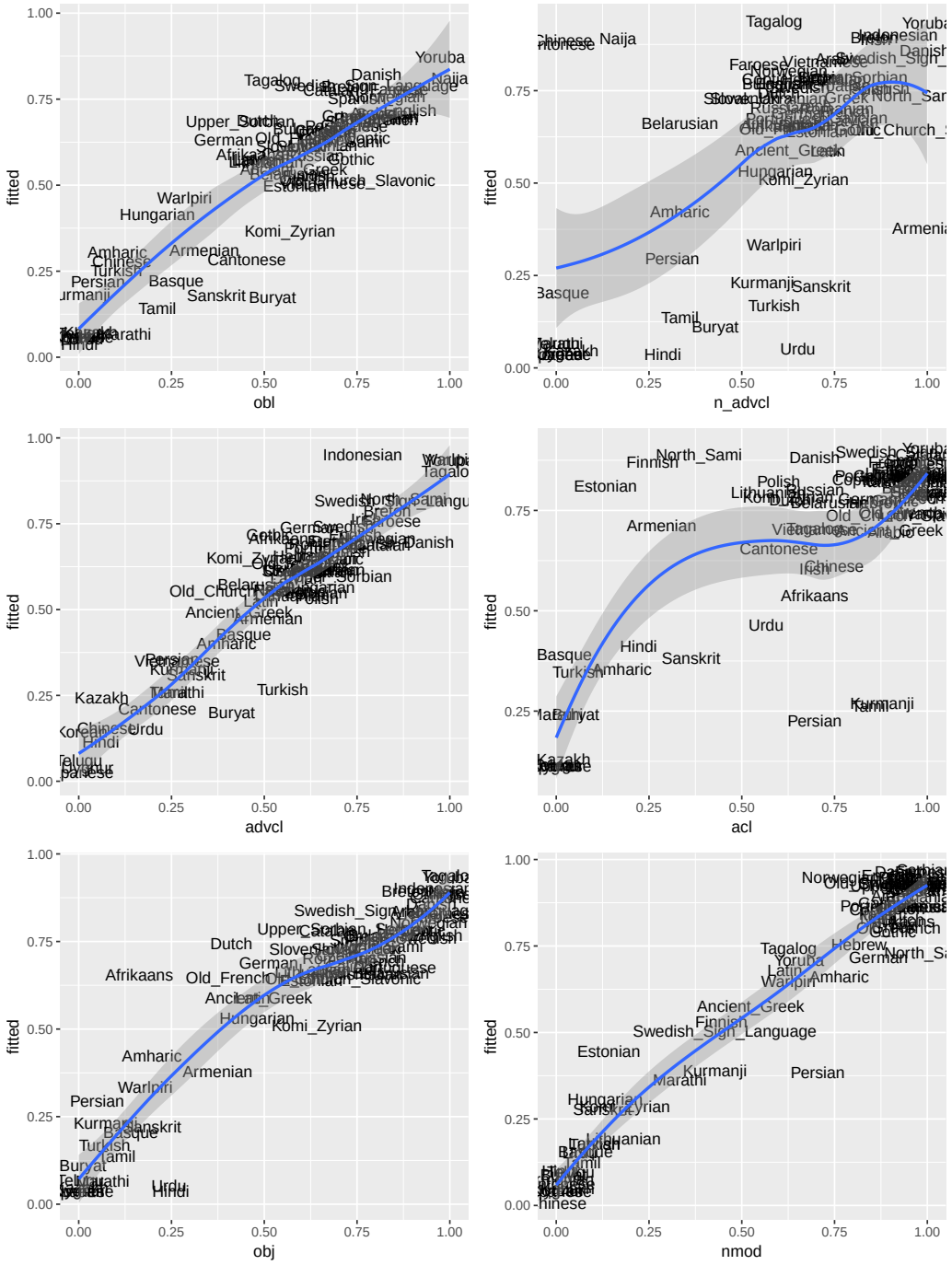


Figure 5: Observed vs fitted predictions for six models. The left column shows verb dependent models for *obl*, *advcl* and *obj*, and the right column shows noun dependent models for (n_)*advcl*, *acl* and *nmod*.

be generalizable to a larger sample.

Secondly, we show that at least some word order universals are not categorical, but in fact gradient. For instance, it is not that OV languages favour postpositions, it is that the proportion of OV vs VO structures in the language correlates with the proportion of post and prepositions. This is a more nuanced claim. As far as we are aware, this is a new observation, and it may help us to rethink the explanations for these phenomena.

A possibility for future work is to distinguish between main and subordinate clauses. We know that word order can vary between main and subordinate sentences, with the later often having a stricter word order like in German or French (Bybee, 2002). Similarly, we could try to distinguish between different types of noun phrases (nominal vs. pronominal), noun phrases of different lengths, and different elements within noun phrases. Also, even though we saw robust crosslinguistic trends such as the relative independence of the position of subjects, some languages of our sample are usually considered to be VSO languages. A more detailed look at specific languages or subfamilies for certain head-dependent orders could show to what extend this corpus is in line with previous language-specific word order observations. To avoid the bias towards Indo-European languages, it might also be helpful to exclude these from the sample and see if the results still hold for the Non-Indo-European subset of the UD treebanks.

Another possible path to take in future work is to try and convert the UD treebanks into different annotation schemes. There is work on converting dependency treebanks into LFG representations (Haug, 2012), for example. If one could convert the UD dependencies into some other theory, this might provide us with structures that make it possible to explore other relations crosslinguistically.

References

Aristar, A. R. (1991). On Diachronic Sources and Synchronic Pattern: An Investigation into the Origin of Linguistic Universals. *Language*, 67(1):1–33.

Arnold, J. E., Losongco, A., Wasow, T., and Ginstrom, R. (2000). Heaviness vs. newness: The effects of structural complexity and discourse status on constituent ordering. *Language*, 76(1):28–55.

Bickel, B. (2008). A refined sampling procedure for genealogical control. *Sprachtypologie und Universalienforschung*, 61:221–233.

Bybee, J. (1988). The diachronic dimension in explanation. In *Explaining Language Universals*, pages 350–379. Blackwell, Oxford.

Bybee, J. L. (2002). Main clauses are innovative, subordinate clauses are conservative: Consequences for the nature of constructions. In Bybee, J. L. and Noonan, M., editors, *Complex Sentences in Grammar and Discourse Essays in Honor of Sandra A. Thompson*, pages 1–18. Benjamins, Amsterdam.

Celano, G. G. A. (2014). A computational study on preverbal and postverbal accusative object nouns and pronouns in Ancient Greek. *The Prague Bulletin of Mathematical Linguistics*, 101(1):97–110.

Chen, X. and Gerdes, K. (2017). Classifying languages by dependency structure. Typologies of delexicalized universal dependency treebanks. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017), September 18-20, 2017, Università Di Pisa, Italy*, pages 54–63.

Cristofaro, S. (2018). Processing explanations of word order universals and diachrony: Relative clause order and possessor order.

Dryer, M. S. (1989). Article-noun order. *Chicago Linguistic Society*, 25:83–97.

Dryer, M. S. (1991). SVO languages and the OV : VO typology. *Journal of Linguistics*, 27(2):443–482.

Dryer, M. S. (1992). The Greenbergian word order correlations. *Language*, 68(1):81–138.

Dryer, M. S. (2009). The branching direction theory of word order correlations revisited. In Scalise, S., Magni, E., and Bisetto, A., editors, *Universals of Language Today*, Studies in Natural Language and Linguistic Theory, pages 185–207. Springer, Dordrecht.

Dryer, M. S. (2019). On the order of demonstrative, numeral, adjective and noun. *Language*.

Dunn, M., Greenhill, S. J., Levinson, S. C., and Gray, R. D. (2011). Evolved structure of language shows lineage-specific trends in word-order universals. *Nature*, 473(7345):79–82.

Futrell, R., Mahowald, K., and Gibson, E. (2015). Quantifying word order freedom in dependency corpora. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 91–100, Uppsala.

Greenberg, J. H., editor (1963). *Universals of Language*. MIT Press, Cambridge, MA.

Gulordava, K. (2018). *Word Order Variation and Dependency Length Minimisation: A Cross-Linguistic Computational Approach*. PhD thesis, University of Geneva.

Gundel, J. K. (1988). Universals of topic-comment structure. In Hammond, M., Moravcsik, E. A., and Wirth, J., editors, *Studies in Syntactic Typology*, pages 209–239. Benjamins, Amsterdam.

Haug, D. T. T. (2012). From dependency structures to LFG representations. In *Proceedings of the LFG12 Conference*, pages 271–291.

Hawkins, J. A. (1979). Implicational universals as predictors of word order change. *Language*, 55(3):618–648.

Hawkins, J. A. (1980). On implicational and distributional universals of word order. *Journal of Linguistics*, 16(2):193–235.

Hawkins, J. A. (1983). *Word Order Universals and Their Explanation*. Academic Press, New York.

Hawkins, J. A. (1994). *A Performance Theory of Order and Constituency*. Cambridge University Press, Cambridge.

Hawkins, J. A. (2014). *Cross-Linguistic Variation and Efficiency*.

Himmelmann, N. P. (1997). *Deiktikon, Artikel, Nominalphrase: Zur Emergenz Syntaktischer Struktur*. Niemeyer, Tübingen.

Junge, B., Theakston, A. L., and Lieven, E. (2015). Given–new/new–given? Children's sensitivity to the ordering of information in complex sentences. *Applied Psycholinguistics*, 36(3):589–612.

Lambrecht, K. (1994). *Information Structure and Sentence Form: Topic, Focus, and the Mental Representations of Discourse Referents*. Cambridge University Press, Cambridge.

Lehmann, W. P. (1973). A structural principle of language and its implications. *Language*, 49(1):47–66.

Lehmann, W. P. (1974). *Proto-Indo-European Syntax*. University of Texas Press, Austin.

Liu, H. (2010). Dependency direction as a means of word-order typology: A method based on dependency treebanks. *Lingua*, 120(6):1567–1578.

Mithun, M. (1992). Is basic word order universal? In Payne, D. L., editor, *Pragmatics of Word Order Flexibility*, pages 15–61. Benjamins, Amsterdam.

Nakagawa, S., Johnson, P. C., and Schielzeth, H. (2017). The coefficient of determination R2 and intra-class correlation coefficient from generalized linear mixed-effects models revisited and expanded. *Journal of the Royal Society Interface*, 14(134).

Nakagawa, S. and Schielzeth, H. (2013). A general and simple method for obtaining R2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2):133–142.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R. T., Petrov, S., Pyysalo, S., Silveira, N., and others (2016). Universal Dependencies v1: A Multilingual Treebank Collection. In *LREC*.

Payne, D. L., editor (1992). *Pragmatics of Word Order Flexibility*. Benjamins, Amsterdam.

Siewierska, A. (1988). *Word Order Rules*. Croom Helm, London.

Smithson, M. and Verkuilen, J. (2006). A better lemon squeezer? Maximum-likelihood regression with beta-distributed dependent variables. *Psychological methods*, 11(1):54.

Song, J. J. (2009). Word order patterns and principles: An overview. *Language and Linguistics Compass*, 3(5):1328–1341.

Taboada, M. and Wiesemann, L. (2010). Subjects and topics in conversation. *Journal of Pragmatics*, 42(7):1816–1828.

Vennemann, T. (1974). Topics, subjects and word order: From SXV to SVX via TVX. In Anderson, J. and Jones, C., editors, *Proceedings of the First International Congress of Historical Linguistics, Edinburgh, September 1973*, pages 339–376. North-Holland, Amsterdam.

Vennemann, T. (1975). An explanation of drift. In Li, C. N., editor, *Word Order and Word Order Change*, pages 269–305. University of Texas Press, Austin, TX.

Universal dependencies and non-native Czech

Jirka Hana, Barbora Hladká

Charles University, Malostranské nám. 25, 118 00 Prague 1, Czech Republic

{hana,hladka}@ufal.mff.cuni.cz

ABSTRACT

CzeSL is a learner corpus of texts produced by non-native speakers of Czech. Such corpora are a great source of information about specific features of learners' language, helping language teachers and researchers in the area of second language acquisition. In our project, we have focused on syntactic annotation of the non-native text within the framework of Universal Dependencies. As far as we know, this is a first project annotating a richly inflectional non-native language. Our ideal goal has been to annotate according to the non-native grammar in the mind of the author, not according to the standard grammar. However, this brings many challenges. First, we do not have enough data to get reliable insights into the grammar of each author. Second, many phenomena are far more complicated than they are in native languages. We believe that the most important result of this project is not the actual annotation, but the guidelines and principles that can be used as a basis for other non-native languages.

KEYWORDS: learner corpus, second language, syntax annotation, universal dependencies, second language acquisition.

1 Introduction

Universal Dependencies (UD) is a unified approach to grammatical annotation that is consistent across languages.[1] It facilitates both linguistic and NLP research. However, the absolute majority of these treebanks are based on corpora of standard language. In this paper, we describe a project of creating a syntactically annotated corpus of learner Czech, the CzeSL corpus. The choice of Universal Dependencies as the annotation standard was relatively straightforward. It is an established framework used for more than 100 treebanks in 60 languages (including two other learner corpora). The common guidelines make the data easily accessible to a large audience of researchers and comparable across languages. Also, following the UD schema and format makes it easier to train and test NLP tools on the basis of our annotation.

CzeSL (Hana et al., 2010), (Rosen et al., 2014) is a learner corpus of texts produced by non-native speakers of Czech.[2] Such corpora are a great source of information about specific features of learners' language, helping language teachers and researchers in the area of second language acquisition. Each sentence in the CzeSL corpus has an error annotation and a target hypothesis with its morphological and syntactic annotation. However, there is no linguistic annotation of the original text. This means we can see what grammatical constructions the authors should have used but not what they actually used. And we can analyze their grammar only indirectly via the error annotation. Therefore we have focused on syntactic annotation of the non-native text within the framework of UD. Figure 1 shows a UD tree structure for (1) selected from CzeSL.

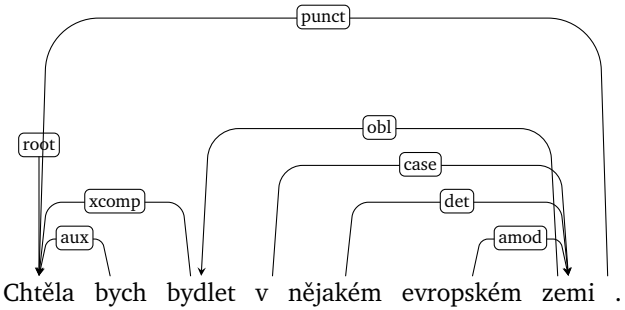


Figure 1: A sample UD tree

(1) Chtěla bych bydlet v nějakém evropském zemi .
I-liked would to-live in some$_{masc}$ European$_{masc}$ country$_{fem}$.
'I would like to live in some European country.'

The remainder of this paper is divided into five sections. In Section 2, we provide an overview of works related to the UD annotation of learner corpora. Section 3 gives a general description of the CzeSL corpus that we annotate in the UD framework. Section 4 presents a core part of the paper. We formulate our annotation principles and describe the challenges that we meet while applying the existing guidelines. The annotation procedure itself is presented in Section 5. In Setion 6, we provide the conclusions.

[1] http://universaldependencies.org
[2] http://utkl.ff.cuni.cz/learncorp/

CORPUS	LANGUAGE	SIZE (annotated)	
TLE (Berzak et al., 2016)	English	5,124 sentences	97,681 words
REALEC (Kuzmenko and Kutuzov, 2014)	English	373 sentences	7,196 words
Tweebank (Liu et al., 2018)	English	3,550 tweets	45,607 words
CFL (Lee et al., 2017)	Chinese	451 sentences	7,256 words

Table 1: Relevant UD-annotated Corpora

2 Related work

The great majority of currently available UD treebanks were converted from already existing treebanks annotated using a different annotation scheme, Moreover, these corpora contain texts written completely by native speakers. The UD annotation of learner corpora has been initiated later on.

Table 1 summarizes UD-annotated corpora relevant for our task. The Treebank of Learner English (TLE) contains manually annotated POS tags and UD trees for sentences selected from the Cambridge First Certificate in English learner corpus (Yannakoudakis et al., 2011). The REALEC corpus is a collection of English texts written by Russian-speaking students. Unlike TLE, the REALEC sample was first automatically annotated by the UDPipe pipeline (Straka and Straková, 2017) and then manually corrected. (Lyashevskaya and Panteleeva, 2018) analyzed the errors made by the parser that originate from differences between English and Russian, typologically different languages. (Lee et al., 2017) have adapted existing UD guidelines for Mandarin Chinese to annotate learner Chinese texts. As an annotation workbench, they used essays written by Chinese language learners representing 46 different mother tongue languages (Lee et al., 2016). As far as we know, there is no similar project for a richly inflected language. We include Tweebank, a twitter corpus, because even though it is not a corpus of non-native language, it brings similar challenges. The wordings and language style used in tweets are often far from the straightforward and well researched syntactic constructions used by the news corpora.

Among UD languages, Czech has an exceptional status because of the greatest number of Czech sentences annotated in UD, namely 127 thousand sentences included in 5 treebanks. Most of the Czech UD treebanks were originally annotated according to the Prague Dependency Treebank annotation scheme[3] and then transformed into UD. The only treebank annotated from scratch is the Czech part[4] of the Parallel Universal Dependencies treebanks created for the CoNLL 2017 shared task (Zeman et al., 2017). Czech holds its exceptional status among the UD treebanks of Slavic languages as well, see (Lasota, Brielen Madureira, 2018). (Zeman, 2015) focuses on a few morphological and syntactic phenomena that occur in Slavic languages and their treatment

[3] https://ufal.mff.cuni.cz/prague-dependency-treebank

[4] https://github.com/UniversalDependencies/UD_Czech-PUD/blob/master/cs_pud-ud-test.conllu

in UD.

The task of corpus annotation deals with a fundamental issue of consistent annotation of the same phenomena within and across corpora. (de Marneffe et al., 2017) assessed the consistency within the Universal Dependency Corpora of English, French, and Finnish by checking dependency labels of words occurring in the same context. (Ragheb and Dickinson, 2013) reports on a study of inter-annotator agreement for a dependency annotation scheme designed for learner English.

3 The CzeSL corpus

The whole CzeSL corpus contains about 1.1 million tokens in 8,600 documents and is compiled from texts written by students of Czech as a second or foreign language at all levels of proficiency. CzeSL-MAN is a subset of CzeSL, manually annotated for errors.[5] It consists of 128 thousand tokens in 645 documents written by native speakers of 32 different languages. In the rest of this paper, when we refer to CzeSL, we refer to CzeSL-MAN. Each CzeSL document is accompanied with:

- metadata – information about the native language of the author, length of study, type of task, etc.
- error annotation (see below)
- linguistic annotation of the target hypothesis

The CzeSL error annotation consists of three tiers:

- Tier 0 (T0): an anonymized transcript of the hand-written original with some properties of the manuscript preserved (variants, illegible strings),
- Tier 1 (T1): forms that are incorrect in isolation are fixed. The result is a string consisting of correct Czech forms, even though the sentence may not be correct as a whole
- Tier 2 (T2): the remaining error types (valency, agreement, word order, etc.), i.e. this is the target hypothesis.

Links between the tiers allow capturing errors in word order and complex discontinuous expressions. Errors are not only corrected, but also classified according to a taxonomy. As an example consider (2) – showing the original text (T0) and the target hypothesis (T2). The full error analysis, including error tags is in Figure 2

(2) T0: Myslím že kdy by byl se svím dítem ...
T2: Myslím , že kdybych byl se svým dítětem ...
think$_{1sg}$, that if-would$_{1sg}$ was$_{masc}$ with my$_{neut.sg.inst}$ child$_{neut.sg.inst}$ ...
'I think that if I were with my child ...'

[5] https://bitbucket.org/czesl/czesl-man/

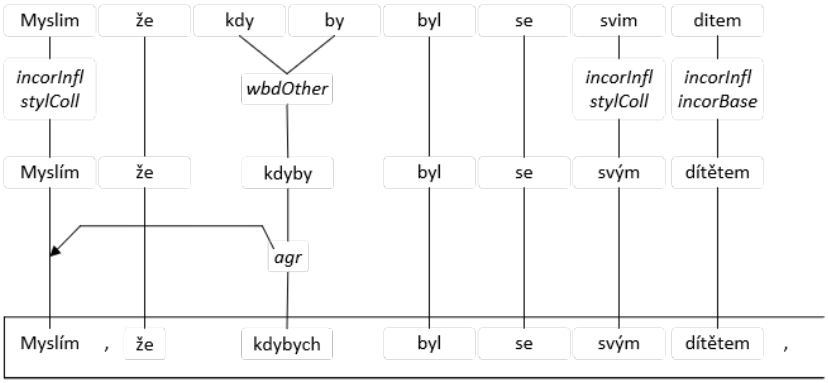


Figure 2: Error annotation of a sample sentence in (2)

Annotation of this kind is supplemented by a formal classification, e.g. an error in morphology can also be specified as being manifested by a missing diacritic or a wrong consonant change. The annotation scheme was tested in two rounds, each time on a doubly-annotated sample – first on a pilot annotation of approx. 10,000 words and later on nearly half of all the data, both with fair inter-annotator agreement results. Error annotation of this kind is a challenging task, even more so for a language such as Czech, with its rich inflection, derivation, agreement, and a largely information structure-driven constituent order.

In addition to error annotation, the target hypothesis is annotated linguistically: for morphology and syntax. However, as mentioned above, there is no linguistic annotation of the original text, a gap we are in a process of filling.

4 Approach

Similarly as the projects above, we follow the basic annotation principle of the SALLE project (Dickinson and Ragheb, 2013), and attempt to annotate literally: we annotate the sentences as they are written, not as they should be. In other words, our ideal goal is to annotate according to the non-native grammar in the mind of the author (i.e. the grammar of their interlanguage), not according to the standard grammar.

However, this brings several challenges. First, in many cases, we do not have enough data to get reliable insights into the grammar of each author. Second, many phenomena are far more complicated than they are in native languages. Our annotation principles include:

- When form and function clash, form is considered less important. For example, if a word functions as an adjective, we annotate it as an adjective even if it has a verbal ending.
- When lacking information, we make conservative statements.
- We focus on syntactic structure and the most important grammatical functions, annotating unclear functions with an underspecified label.

4.1 Tokenization

There is an established tokenization used by Czech UD corpora that builds on the general UD tokenization rules. However, we used the original CzeSL tokenization to make the UD structures compatibles with its error annotation. The differences affect mostly alternatives offered by the author due to their uncertainty (e.g. *b(y/i)l* is considered one token), hyphenated words and certain numerical expressions.

4.2 Part-of-speech and Morphology

Czech, as other Slavic languages, is richly inflected. It has 7 cases, 4 genders, colloquial variants, etc. Therefore, corpora of standard Czech are usually annotated with detailed morphological tags (for example, the tagset used for the Prague Dependency Treebank has 4000+ tags, distinguishing roughly 12 different categories). We have decided not to perform such annotation. There are several reasons, for this decision, mainly:

- many endings are homonymous; therefore it is not obvious which form was used if we wanted to annotated according to the form. For example, the ending -a has more than 10 different morphological functions depending on the paradigm.
- these complications do not always correlate with understandability. Some texts are easy to understand yet, they use wrong or non-existing suffixes, mix morphological paradigms etc.
- the corpus can be still searched for pedagogical reasons: the intended morphological tag can be derived from the corresponding target hypothesis, the error annotation marks mistakes in inflection and the original forms can be matched existing standard forms

Instead, we have limited ourselves to the Universal POS Tagset (Petrov et al., 2011). When form and function clash, form is considered less important. For example, if a word functions as an adjective, we annotate it as an adjective even if it has a verbal ending.

One of the common deviating characteristics of learner Czech was the neutralization between adjectives and adverbs. In (3), the adjective *rychlé* 'quick' is used instead of the correct adverb *rychle* 'quickly'.

(3) T0: Kvalita života by se zlepšila moc rychlé.
T2: Kvalita života by se zlepšila moc rychle.
Quality of-life would refl improve too quick(ly)
'Life quality would improve too quickly.'

This is similar to German or colloquial English. Unfortunately, UPOS force us to choose between adjectives and adverbs even for speakers who clearly use the same word for both. We annotate such words as adjectives with an additional note.

4.3 Lemmata

Ideally, we would use lemmata from the author's interlanguage. For example, in (4), we would use the lemma *Praga* (correctly *Praha*). The situation is clear, because the word is in the lemma form already (nominative singular). Often knowing the native language of the author helps – for example, in (5) the lemma of *krasivaja* is *krasivyj*, based on Russian.

(4) T0: Praga je hezké město. → lemma: Praga
T2: Praha je hezké město. → lemma: Praha
Prague is nice city
'Prague is a nice city.'

(5) T0: Praga je krasivaja. → lemma: krasivyj
T2: Praha je krásná. → lemma: krásný
Prague is beautiful
'Prague is beautiful.'

Sometimes we can see that the author declines a word using a paradigm of another word. For example, for *večeřem* 'dinner$_{inst}$ in (6) we can hypothesize the masculine lemma *večeř*, formed in analogy with the word *oběd – obědem* 'lunch'. The correct forms are feminine *večeře – večeří*.

(6) T0: Začíname večeřem. → lemma: večeř
T2: Začíname večeří. → lemma: večeře
we-start with-dinner$_{inst}$
'We start with dinner.'

However in many cases, the situation is much more complicated and it is not clear whether a certain deviation is due to a spelling error, incorrect case (Czech has 7 cases + prepositions), wrong paradigm (Czech has 14+ basic noun paradigms) or simply a random error. Sometimes, we can see particular patterns in the whole document, e.g. the author uses only certain cases, or certain spelling convention (Russian speakers sometimes use '*g*' instead of Czech '*h*'), not distinguishing between adjectives and adverbs, etc. These patterns can help us to deduce lemmas in concrete cases. Unfortunately, in some cases we simply do not have enough data to reliably deduce the correct lemma. In that case, we are trying to be as conservative as possible and assume as little as possible: we use the form of the word as its lemma and mark it as unclear in the note field.

The alternative is to use the correct lemma (*Praha* in (4) and *večeře* in (6)). This would obviously make the situation clearer and the annotation more reliable. However, the benefit would be minimal: error annotation already provides us with the correct forms so we can easily derive their lemmas using available approaches for standard native language.

4.4 Syntactic Structure

In annotating syntactic structure, we again follow the rule of annotation the structure of interlanguage. For example, if the learner uses the phrase (7), the word *místnost* 'room' is annotated as a direct object (OBJ), even though a native speaker would use an adverbial (OBL) *do místnosti* 'into room' as in (8).

(7) vstoupit místnost$_{OBJ}$
enter room
intended: 'enter a/the room'

(8) vstoupit do místnost$_{OBL}$
enter into room
'enter a/the room'

5 Annotation procedure

For a pilot annotation, we have randomly selected 100 sentences shorter than 15 tokens. The average sentence length is 6.8. Technically, we use the TrEd editor with the *ud* extension to display and edit Universal Dependency trees and labels.[6]

An annotator with a philological background and a secondary-school student annotated the sample. They did not annotate the sentences from scratch, but corrected the output of UDPipe (Straka and Straková, 2017). They did not undergo any special training prior to the annotation, but instead relied on a secondary-school grammar training and the guidelines for Czech available at the UD project site.[7] When they were not sure with a particular construction, they referred to existing Czech and English UD corpora, compiling shared guide and a cheat sheet[8] in the process.

UPOS	LABEL	REL
0.934	0.89	0.927

Table 2: Inter-annotator agreement on the sample of CzeSL measured using Cohen's *kappa* on UPOS labels, syntactic labels and unlabelled heads respectively

6 Conclusion

We are in the process of creating a syntactically annotated corpus of learner Czech. So far, we have annotated around 2,000 sentences. The goal is to annotate all of the approximately 11 thousand sentences in CzeSL. To the best of our knowledge this is a first such corpus of any inflectional language. We are also planning to have a significant portion of the corpus annotated by two annotators. Currently, we have only around 100 sentences doubly annotated with a good but not perfect inter-annotator agreement. We believe that the most important result of this project is not the actual annotation, but the guidelines that can be used as a basis for other non-native languages. The high-level annotation principles of ours include: (1) When form and function clash, form is considered less important. (2) When lacking information, we make conservative statements. (3) We focus on syntactic structure and the most important grammatical functions, annotating unclear functions with an underspecified label.

Acknowledgments

We thank Filip Hana and Jana Vitoušová Alferyová for annotating the data and commenting the annotation guidelines. We gratefully acknowledge support from the Grant Agency of the Czech Republic, grant No. ID 16-10185S.

[6] `https://ufal.mff.cuni.cz/tred`
[7] `http://universaldependencies.org/guidelines.html`
[8] `http://bit.ly/UDCheat`

References

Berzak, Y., Kenney, J., Spadine, C., Wang, J. X., Lam, L., Mori, K. S., Garza, S., and Katz, B. (2016). Universal dependencies for learner english. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 737–746. Association for Computational Linguistics.

de Marneffe, M.-C., Grioni, M., Kanerva, J., and Ginter, F. (2017). Assessing the annotation consistency of the universal dependencies corpora. In *Proceedings of the Fourth International Conference on Dependency Linguistics (DepLing)*, pages 108–115, Pisa, Italy.

Dickinson, M. and Ragheb, M. (2013). Annotation for learner english guidelines, v. 0.1 (june 2013).

Hana, J., Rosen, A., Škodová, S., and Štindlová, B. (2010). Error-tagged Learner Corpus of Czech. In *Proceedings of The Fourth Linguistic Annotation Workshop (LAW IV)*, Uppsala.

Kuzmenko, E. and Kutuzov, A. (2014). Russian error-annotated learner english corpus: a tool for computer-assisted language learning. In *Proceedings of the third workshop on NLP for computer-assisted language learning at SLTC 2014, Uppsala University*, number 107, page 87–97. Linköping University Electronic Press, Linköpings universitet.

Lasota, Brielen Madureira (2018). Slavic Languages and the Universal Dependencies Project: a seminar. `http://www.coli.uni-saarland.de/~andreeva/Courses/SS2018/SlavSpr/presentation_25062018).pdf`. 25 June 2018.

Lee, J., Leung, H., and Li, K. (2017). Towards universal dependencies for learner chinese. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 67–71. Association for Computational Linguistics.

Lee, L.-H., Chang, L.-P., and Tseng, Y.-H. (2016). Developing learner corpus annotation for chinese grammatical errors. *2016 International Conference on Asian Language Processing (IALP)*, pages 254–257.

Liu, Y., Zhu, Y., Che, W., Qin, B., Schneider, N., and Smith, N. A. (2018). Parsing tweets into universal dependencies. *CoRR*, abs/1804.08228.

Lyashevskaya, O. and Panteleeva, I. (2018). REALEC learner treebank: annotation principles and evaluation of automatic parsing. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 80–87, Prague, Czech Republic.

Petrov, S., Das, D., and McDonald, R. T. (2011). A universal part-of-speech tagset. *CoRR*, abs/1104.2086.

Ragheb, M. and Dickinson, M. (2013). Inter-annotator agreement for dependency annotation of learner language. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 169–179, Atlanta, Georgia. Association for Computational Linguistics.

Rosen, A., Hana, J., Štindlová, B., and Feldman, A. (2014). Evaluating and automating the annotation of a learner corpus. *Language Resources and Evaliation*, 48(1):65–92.

Straka, M. and Straková, J. (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, Vancouver, Canada. Association for Computational Linguistics.

Yannakoudakis, H., Briscoe, T., and Medlock, B. (2011). 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 - Volume 1*, HLT '11, pages 180–189, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zeman, D. (2015). Slavic languages in universal dependencies. In Gajdošová, K. and Žáková, A., editors, *Natural Language Processing, Corpus Linguistics, E-learning*, pages 151–163, Lüdenscheid, Germany. Slovenská akadémia vied, RAM-Verlag.

Zeman, D., Popel, M., Straka, M., Hajič, J., and Nivre, J. (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, Stroudsburg, PA, USA. Charles University, Association for Computational Linguistics.

On the Development of a Large Scale Corpus for Native Language Identification

Thomas. G. Hudson, Sardar Jaf

Durham University, Durham, UK

g.t.hudson@durham.ac.uk, sardar.jaf@durham.ac.uk

Abstract

Native Language Identification (NLI) is the task of identifying an author's native language from their writings in a second language. In this paper, we introduce a new corpus (italki), which is larger than the current corpora. It can be used for training machine learning based systems for classifying and identifying the native language of authors of English text. To examine the usefulness of italki, we evaluate it by using it to train and test some of the well performing NLI systems presented in the 2017 NLI shared task. In this paper, we present some aspects of italki. We show the impact of the variation of italki's training dataset size of some languages on systems performance. From our empirical finding, we highlight the potential of italki as a large scale corpus for training machine learning classifiers for classifying the native language of authors from their written English text. We obtained promising results that show the potential of italki to improve the performance of current NLI systems. More importantly, we found that training the current NLI systems on italki generalize better than training them on the current corpora.

Keywords: Native Language, training data, italki, NLI, Native Language Identification, Language Identification, Dataset, Corpus.

1 Introduction

In the modern world where English is commonly used as the lingua franca of business and commerce, non-native speakers vastly outnumber native speakers of English. The study of the way non-native speakers of a language learn a new language is known as Second Language Acquisition (SLA), which focuses on the influence of the speaker's first language (FL) on their second language (SL) (Jarvis & Crossley 2012). Two types of analyses are considered in SLA: detection and comparison. The detection-based approach involves the use of large amounts of data to identify subtle patterns in the way SL usage differs from FL usage. The comparison-based approach is often used by linguists. It involves studying the differences between FLs to form hypotheses of the way these differences impact the speaker's use of their acquired language (SL).

The rise of ubiquitous computing and the availability of vast amount of data on the Internet has led to the popularity of the detection-based approach, which leads to the computational linguistics problem of native language identification.

Native language identification (NLI) is the process of identifying a writer's FL from a text (or speech). Computationally, it is a classification task. Supervised learning is the most popular approach to identifying the native language of an author from a text.

Solutions to this task have many applications. Analysis of successful systems provide insight into the theoretical linguistics, which underpins second language acquisition, types of text features that are successful are likely to emerge from some aspects of a group of FLs, which can then be examined further. Real world applications such as language teaching can be improved by identifying common mistakes to indicate areas of difficulty tailored to specific FL backgrounds (Rozovskaya & Roth 2011). NLI also has applications in forensic linguistics as part of wider studies on author profiling (Estival et al. 2007, Gibbons 2003) (e.g. Intelligence/security agencies can use NLI systems to build a profile of their target/suspect).

In the paper we provide details of the process for development a large scale dataset for NLI, empirical analyses of using the proposed dataset for the task of NLI and our future direction for further enhancing the proposed dataset . In section 2 we outline the related work. In section 3 we highlight various aspects of the proposed corpus and draw detailed comparison between the proposed corpus and the current corpora. In section 4 we evaluate the usefulness of the corpus for NLI task. Finally, section 5 concludes our work and set out our future work.

2 Related Work

Text classification systems (including solutions for native language identification) usually follow a structure outlined in Figure 1, which consists of four separate components.

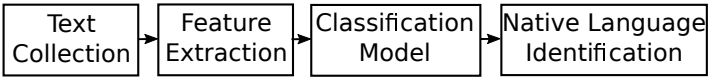


Figure 1: Framework of Tofighi et al. (2012).

Text collection involves collecting a body of text (corpus) for the task. Many approaches use pre-existing corpora made with the NLI task in mind. Next, since supervised approach is usually used for NLI, features (which are often in raw text format) are extracted from the corpus and converted to numerical attributes. These features are then supplied to a classification model, which utilizes machine learning algorithms to learn patterns and to distinguish between labelled data. Finally, the

model is used to perform native language identification on unseen texts (i.e., identifying the label (FL) of the text).

The main two research events in this area were the shared task 2013 on native language identification (Tetreault et al. 2013), and more recently in 2017 (Malmasi et al. 2017). These landmark events consist of multiple teams competing to advance the state-of-the-art by designing systems to solve the task on a single shared dataset to allow a direct comparison between all approaches. The winning approaches in these tasks have become highly influential in the design of subsequent solutions to NLI.

All known solutions to the NLI problem use supervised learning. Thus, they rely on a collection of data (corpus) labelled with the native language of the author in order to train machine learning classifiers. In 2002, Granger et al. (2002) introduced the International Corpus of Learner English (ICLE) corpus to be used for the task of NLI. ICLE consists of argumentative essays written by university students of English in response to a range of prompts. Each essay is associated with a learner profile and it includes various aspects of the writer such as author's age, gender and native language.

The main limitation of ICLE, as argued by Tofighi et al. (2012), is its heavy topic bias and its character encoding, which could lead to inflated accuracy on ICLE. These issues prevent machine learning classifiers trained on it from generalizing to real-world examples, and possibly causing the conflation of native language identification and topic identification– another task in natural language processing.

These issues led Tetreault et al. (2012) to the introduction of the 'The Test of English as a Foreign Language (TOEFL)' corpus for the first NLI shared task in 2013. TOEFL has since become the de-facto standard in NLI studies with an updated version being released for the most recent NLI shared task in 2017 (Malmasi et al. 2017). The TOEFL corpus was designed to mitigate problems of topic bias, which plagued earlier corpora, by designing the collection process to sample prompts equally across all FLs. Refined for use in the 2017 shared task, it is the standard benchmark for comparing NLI systems.

A key problem with these existing corpora is their limited size - TOEFL only provides 1000 documents per language. While the current corpus size has been sufficient for training many shallow learning algorithms (e.g., Support Vector Machines which helped many systems to obtain impressive performance) it is not sufficient for most deep neural network algorithms (which often require data sizes many orders of magnitude larger than the TOEFL dataset). Dramatically increasing the size of the dataset could significantly improve the accuracy of NLI systems and allow deep neural network algorithms to contribute to this task, as it has contributed to many natural language processing (NLP) tasks.

There have been attempts to increase dataset size. Brooke & Hirst (2012) attempted to increase the dataset size in their studies on NLI by using 2 stages of machine translation (SL→FL→SL) on a large English corpus. One of the advantage of this approach is it helps in generating a very large corpus for NLI, where features of the FL are transferred. Another advantage is the removal of any possibility of topic bias as the source documents are all randomly sampled from the same corpus. However, empirically, this produced poor but above baseline results as it eliminated more nuanced features, such as misspelling, which are commonly used to boost the accuracy of state-of-the-art systems.

An alternative approach to Brooke & Hirst (2012) is web-scraping, which is a common practice

in many areas of NLP. Web-scraping can provide diverse and vast quantities of data (big data) especially important for training deep neural networks (deep learning). As some datasets (which are many times larger than TOEFL) for deep learning show promise in other areas of NLP . The introduction of a new large corpus for NLI, which could be based on the italki website[1], should improve the performance of current shallow learning based systems, and enable deep learning to algorithm to contribute to the NLI problem.

The available corpora have been used for training various learning classifiers. The most widely used classifier in NLI has been linear Support Vector Machines (SVMs) which were used by Koppel et al. (2005) in their seminal work and by the winners of both NLI shared tasks 2013 (Tetreault et al. 2013) and 2017 (Malmasi et al. 2017)

k-Nearest Neighbors and MaxEnt classifiers that have been used in the NLI task yielded a performance competitive to SVMs (Tetreault et al. 2013). Introduced by Tetreault et al. (2012), the use of multiple classifiers has become a key component in many state-of-the-art systems for NLI.

Deep learning algorithms have been tried as an alternative to shallow learning approaches in the 2017 NLI shared task (Malmasi et al. 2017). Teams experimented with simple neural models, namely multilayer perceptrons on n-gram features, but found that SVMs produce higher accuracy in shorter times (Chan et al. 2017). It is generally accepted that deep learning algorithms require large quantities of data in order to learn representations of data and perform well. We believe that the availability of an open-source large dataset will help the NLI research community to explore the application of deep learning to NLI further.

3 The italki Corpus

There are different issues with the currently available corpora, chiefly the high cost of licenses and corpus size. To address those two issues, we propose a web-based corpus. We have gathered large quantities of text from the language learning website italki. The italki website creates a community for language learners to access teaching resources, practice speaking, discuss topics and ask questions in their target language (the English language). The raw data available on the italki website is in free-form 'Notebook' documents, which are mainly autobiographical diary entries with connected profiles describing the native language of the author. The required text and related metadata are retrieved from the italki website via an application programming interface (API).

For this work, we have gathered data for 11 languages (Arabic (ARA), Chinese (CHI), French (FRE), German (GER), Italian (ITA), Hindi (HIN), Japanese (JPN), Korean (KOR), Telugu (TEL), and Turkish (TUR)). The collected data[2] have undergone a similar normalization process as the TOEFL corpus. In the following subsection, we describe the different normalization processes the proposed corpus undergone.

3.1 Normalization

We follow the same practice for removing noisy features from the italki corpus as used in creating the TOEFL corpus. The noisy features that we have removed are listed below:

- **URL removal**: An analysis of the corpus showed that many authors include URLs in their writing. To prevent possible confusion of the classifier, we defined multiple regular expressions

[1] Available at: www.italki.com

[2] The public repository for the corpus is available at: https://github.com/ghomasHudson/italkiCorpus/releases/tag/v1.0 and https://bitbucket.org/sardarjaf/italki-corpus/downloads/

to remove them. This problem is also somewhat mitigated in the following step.

- **Long word removal**: We remove words that are more than fifteen characters of length. These words usually consist of missed URLs or sequences of control/HTML characters that do not generalize to other, off-line, scenarios.
- **Non-standard spacing removal**: Some authors used multiple spaces between words. These are removed. The above processes resulted in substantially cleaner data. Figure 2 shows an example of a normalized text. In the following subsection, we compare various aspects of the italki corpus against the TOEFL corpus.

> We had power cut until 2 hours ago besides the water cut . Just now weve had both thankfully . Being a crowded family , the absence of any of these creates some problems for us . The stars are shining in the sky now . Our house overlooks a barren valley in which horses , goats or sheep graze sometimes . This pastoral view feels as if I were in a village rather than a city . Theres hardly any rush in the vicinity no matter what time it is . Although our house doesnt possess a spectacular view , I like it for its tranquil Environment . Perhaps this is because of the fact that its one of the houses on the road to a nearby village . Weve been informed long before that power network in our apartment is connected to the nearby villages power network , which accounts for why we become extremely indignant when this power cut repeats . I am looking forward to Friday when waters said to will be provided to our vicinity . Water coming from Water trucks is not quite handy nor incessant since a whole building makes use of it .

Figure 2: Example of normalisation.

3.2 Comparing italki to TOEFL

The result of the post-normalization stage was a large corpus suitable for training machine learning algorithms for classifying text and identifying the native language of authors from their written English text. The data format in italki matches that of the TOEFL dataset. This allows the research community to easily use italki in future work without changing data import routines significantly. The structure of the corpus is shown in Figure 3. The corpus is organized in sub folders. The

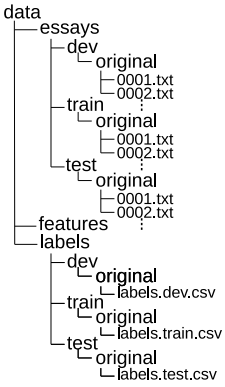


Figure 3: Corpus folder structure.

entire corpus is within the 'data' folder, which contains the essays, features and labels. The data in the corpus was randomized and it was divided to different sections. The essays and labels folders contain subfolders (development (dev), train and test folders). The data in the dev, train and test are used for validation, training and testing machine learning classifiers, respectively. The subfolders in the essays folder contain the raw data in 'txt' file format. The subfolders in the labels folder contain comma separated value (csv) files which have information about the labels (native languages) for

each document in each subfolder in the essays folder. The features folder is empty but can be used to temporarily store features such as part-of-speech tags by NLI systems.

Detailed information about the size of the data in the train, development and test folders for each language is presented in Table 1 The number of documents per language in italki is significantly

Lang	Train			Dev			Test		
	#Docs	#Sent	Words	Docs	Sent	Words	Docs	Sent	Words
Arabic	12035	55117	869008	1281	5759	87858	1465	6346	102175
Chinese	12113	120429	1998294	1348	13557	223239	1530	14815	244141
French	8129	66338	1001566	954	7782	118755	994	8311	127138
German	1563	15496	227172	178	1869	28359	188	1722	25499
Hindi	3942	29790	450335	438	3004	44826	469	3642	53427
Italian	8455	68017	1052870	952	7961	123068	991	7945	122275
Japanese	14327	137946	1648510	1542	15179	178820	1776	17226	205242
Korean	11205	121776	1488519	1239	13142	160926	1389	14604	177479
Spanish	12183	111117	1867441	1325	11947	199401	1006	13658	224963
Telugu	509	2498	32472	57	337	4944	48	229	2910
Turkish	6175	36027	409410	747	4049	48292	875	5049	57732

Table 1: Data sizes for train, development (Dev) and test. Lang=languages, #Docs=number of documents, #Sents=number of sentences and #Words=number of words.

larger than the number of documents per language in TOEFL (which is 1100 documents per language) except for Telugu. The italki corpus contains approximately 122,000 documents in total across a wide range of realistic topic areas, and it contains data for the same languages as in the TOEFL corpus. The content of the documents are raw text written in English by authors whose English is their second language. For each document, the native language of the author is used as the label for that document.

Table 2 shows the data size across the 11 languages in italki and TOEFL corpora. For many of the languages we see 10-fold increase in the document size in italki compared to TOEFL. However, for Telugu, the number of documents in italki is less than that in TOEFL. The number of documents in italki for German, Hindi and Turkish, though it is larger than TOEFL, it is comparatively smaller than for other languages in italki. This creates data imbalance in italki. One of our future goal is to increase the data size for Telugu, German, Hindi and Turkish to eliminate the current data imbalance issue. The total number of words in italki for the majority of the languages is larger than the number of words in TOEFL. italki contains more words for Arabic, Chinese, French, Italian, Japanese, Korean, and Spanish than TOEFL. Hindi and Turkish also has more words in italki than TOEFL. However, there are less words for German in italki than there are in TEOFL. Telugu has exceptionally smaller number of words in italki than in TOEFL.

Despite the fact that italki appears smaller than TOEFL in terms of the average number of words per document/sentence and average sentence length per document, the total number of words and total number of sentences, for most of the languages, is much larger than those in TOEFL (as shown in and Table 2). However, the total number of sentences and the total number of words per language in italki is substantially larger than those in TOEFL for most of the languages, with the exception of Telugu and German, as shown in Table 2.

Language	italki			TOEFL		
	#docs	#sentences	#words	#docs	#sentences	#words
Arabic	14781	67222	1059041	1100	13813	314666
Chinese	14991	148801	2465674	1100	19883	358605
French	10077	82431	1247459	1100	18662	354484
German	1929	19087	281030	1100	19769	381161
Hindi	4849	36436	548588	1100	19620	391970
Italian	10398	83923	1298213	1100	14588	326679
Japanese	17645	170351	2032572	1100	19171	306794
Korean	13833	149522	1826924	1100	20530	332850
Spanish	15003	136722	2291805	1100	15695	363675
Telugu	614	3064	40326	1100	18626	368882
Turkish	7797	45125	515434	1100	19693	352911
Total	111,917	942,684	13,607,066	12,100	200,050	3,852,677

Table 2: Number of documents per language in italki and TOEFL corpora.

There are few major differences between italki and TOEFL. In italki, each document contains 60-150 words per languages. This is significantly different from TOEFL[3], where each document per language contains between 300 and 400 words. Figure 4 shows a comparison between italki and TOEFL with regards to this feature.

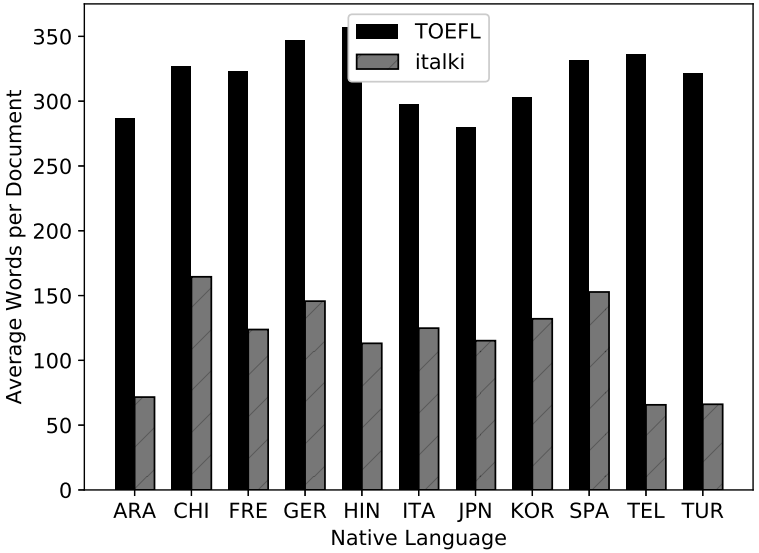


Figure 4: Average words per document for different languages in italki and TOEFL.

To further explore the differences between italki and TOEFL, we compare a range of metrics on the documents. From Figures 5 and 6 we can see that, on average, italki documents contains shorter

[3]This particular difference could make italki more appropriate for training NLI systems targeting short text, such as identifying the native language of authors of short messages.

sentences and fewer sentences than TOEFL, respectively. This reflects the nature of the italki website as a collection of free-form text, often diacritic entries as opposed to the structured essay responses in TOEFL. However, this feature could be useful for training system to perform NLI task on social media data, which often has a similar profile.

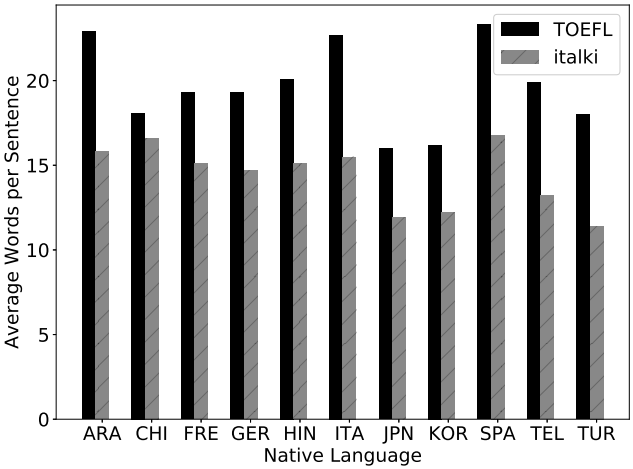


Figure 5: Comparing italki and TOEFL: Average sentence length per document.

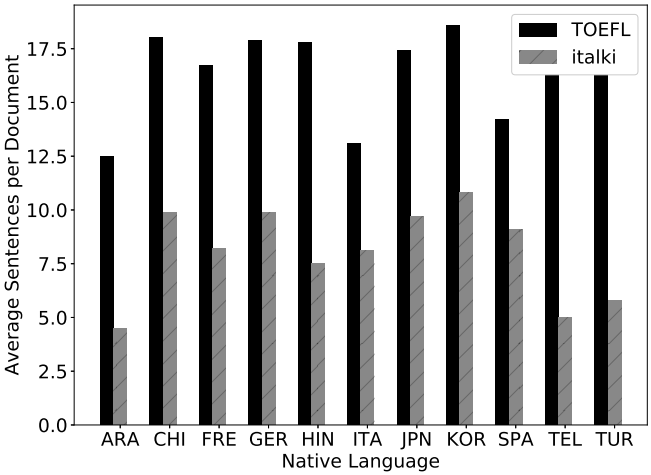


Figure 6: Comparing italki and TOEFL: Average number of sentences per document.

4 Corpus Evaluation

Supervised learning approach is usually used for the NLI problem. For this approach, a corpus of data, labelled with the native language of the author, is required. We evaluate the suitability of italki

for the NLI problem by using it to train and evaluate several existing NLI systems. We choose four systems from the teams in the 2017 shared task (Malmasi et al. 2017) where implementations are readily available:

- **Groningen** (Kulmizev et al. 2017) - Character 1-9-grams classified with a linear SVM.
- **Tubasfs** (Rama & Coltekin 2017) - Character 7-grams and word bigrams classified with a linear SVM.
- **NLI-ISU** (Vajjala & Banerjee 2017) - 1-3-grams classified with MaxEnt, a probabilistic classifier which picks a model based on its entropy (Nigam et al. 1999).
- **Uvic-NLP** (Chan et al. 2017) - Character 4-5-grams and word 1-3-grams classified with a linear SVM.

The performance of several systems in the NLI 2017 shared task, when trained and tested on the TOEFL corpus, is shown in Table 3

System	F1
Groningen	0.8756
Tubasfs	0.8716
Uvic-NLP	0.8633
NLI-ISU	0.8264

Table 3: System performance from NLI 2017 shared task (Malmasi et al. 2017).

These systems were chosen based on their performance ranking (1, 2 and 3) in the NLI 2017 shared task and the retraining and on italki data. In the following subsection, we show our evaluation of those systems when they are trained and tested on the italki corpus.

Following the change in the 2017 shared task, We evaluate and rank the systems identified in Section 4 by their macro F1 score (averaging the per class F1 score across all classes (languages)) rather than accuracy as in earlier studies. We use the definition from Yang & Liu (1999) in order to combine recall (r) and precision (p) over q classes. This ensures systems which perform consistently across all classes are rewarded.

4.1 The Impact of Data Size on System Performance

One of the main advantages of using italki over the current corpora is the data size. In this section we will investigate the impact of a larger web-scraped corpus on the performance of the current, or potential, state-of-the-art systems. For this objective, we explore the affect of different training data sizes on four different NLI systems that were presented in the NLI 2017 shared task(Malmasi et al. 2017).

Experiment #1: NLI systems' performance on italki. For this experiment, we incrementally increase the size of the training set from 10,000 documents across all the eleven languages to 110,000 documents but we keep the test size to 1,000 documents. It is important to note that the training set is not balanced. Some languages have less training data than others. For example, Telugu has just over 500 documents while many other languages have more than 10,000 documents. As presented in Figure 7, we find that increasing the training data size to around 60000 documents it improves the systems' performance linearly. It is worth noting that the systems do not improve when the training size goes beyond 60000 documents. Although this is likely due to the nature of shallow learning algorithms (where even significant increase in the training size does not contribute

to their learning) and the selected systems utilize shallow learning algorithms for training. However, the main reason is most likely due to the data imbalance of italki.

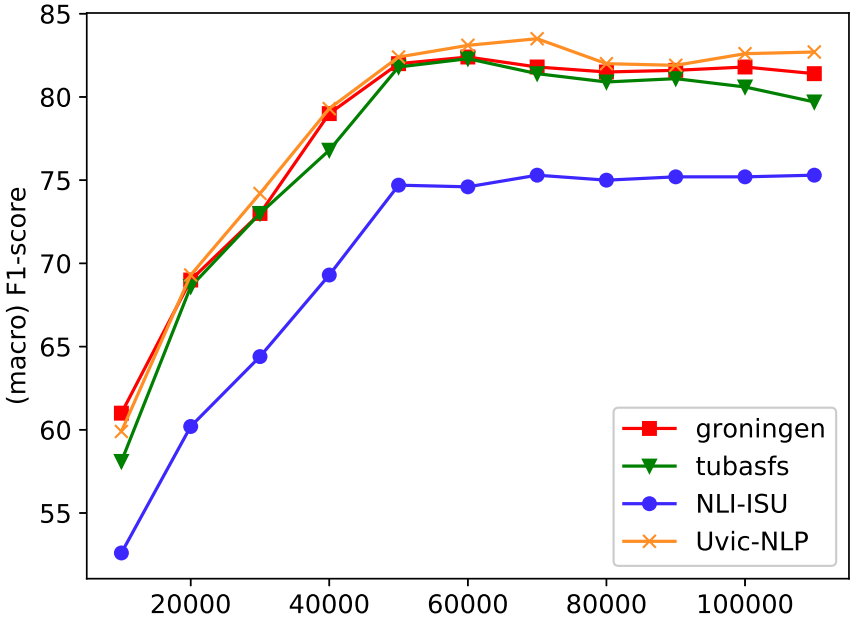


Figure 7: Systems performance when trained on italki corpus.

In the following experiments, we evaluate the performance of Tubasfs by incrementally increasing the size of the training data for each language from 500 to 7000 documents. The main reason for using Tubasfs is due to time constraint. The training time of Tubasfs was significantly shorter than other systems.[4]

Experiment #2: Tubasfs system performance on unbalanced italki. Training one of the top performing systems of the NLI 2017 shared task on italki yield lower accuracy than training them on TOEFL. As we mentioned earlier, currently, italki dataset is not well balanced across all the eleven languages. For a few languages (Telugu, German and Hindi) the training data is much smaller than the available training data for other languages. To understand the impact of the data imbalance of italki on the performance of the NLI systems, we evaluated the Tubasfs system[5] performance for each language by increasing the size of the training data incrementally while keeping the test size (1000 documents) constant as before. Table 4 shows the system's performance on different languages. We have gradually increased the number of documents in the training set from 500 to 7000 documents. The system performance degrades when the training size is increased beyond

[4]We have conducted 27 experiments in total to assess the impact of the data imbalance of italki, hence the use of a fast system for training, such as Tubasfs, was important.

[5]We have not experimented with the other available systems because of the time and space constraint. Our choice of using Tubasfs is mainly because of the speed of training it compared to other systems.

the available training data for some languages (Telugu, German and Hindi). It also appears that the data imbalance affects some of those languages with substantial training data (such as French, Italian and Spanish). The average system performance declines when the training set goes beyond 6000 documents. However, as presented in Figure 7 it appears the system achieves its optimum performance (83%) if the training size is dramatically increased (40,000 document). The impact of the data imbalance is evident in those languages with small training size in the corpus, such as Telugu, as it can be noted from Table 4.

Training Data	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR	AVG
500	0.349	0.573	0.374	0.550	0.776	0.487	0.581	0.535	0.451	0.557	0.360	0.508
1000	0.424	0.581	0.457	0.613	0.888	0.537	0.602	0.577	0.505	0.556	0.395	0.558
1500	0.534	0.599	0.544	0.612	0.900	0.591	0.632	0.617	0.578	0.504	0.429	0.594
2000	0.530	0.644	0.570	0.631	0.818	0.653	0.673	0.616	0.630	0.422	0.834	0.638
2500	0.574	0.693	0.578	0.607	0.786	0.627	0.676	0.624	0.616	0.274	0.875	0.630
3000	0.543	0.689	0.632	0.599	0.780	0.663	0.861	0.698	0.624	0.248	0.916	0.659
4000	0.567	0.708	0.860	0.570	0.747	0.684	0.896	0.705	0.680	0.150	0.946	0.683
5000	0.629	0.772	0.907	0.581	0.744	0.711	0.952	0.908	0.694	0.216	0.960	0.734
6000	0.650	0.765	0.920	0.545	0.762	0.724	0.947	0.930	0.708	0.183	0.946	0.735
7000	0.629	0.751	0.907	0.563	0.770	0.694	0.938	0.938	0.702	0.216	0.914	0.730

Table 4: Systems performance per language when trained on italki.

Experiment #3: Tubasfs system performance on relatively balanced italki. Once we identified the impact of the sparse training set of some languages, such as Telugu, in italki on the system, we conducted further experiment by excluding Telugu from the experiment because its training much smaller than other languages. We also conducted controlled experiment by omitting German and Hindi from this experiment once the training size reached a rate where their performance degraded in the experiment #2 (Hindi at 2000 documents and German at 2500 documents). As shown in Table 5, it can be noted that the system performance improved for all the languages. The average accuracy of the system increased by approximately 13%.

Training Data	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TUR	AVG
1000	0.477	0.583	0.484	0.630	0.917	0.545	0.602	0.593	0.542	0.435	0.581
1500	0.563	0.599	0.555	0.611	0.957	0.602	0.635	0.623	0.583	0.472	0.620
2000	0.579	0.635	0.573	0.641	_	0.649	0.670	0.631	0.638	0.871	0.654
2500	0.627	0.706	0.625	_	_	0.667	0.689	0.660	0.687	0.920	0.698
3000	0.636	0.706	0.667	_	_	0.709	0.861	0.731	0.716	0.956	0.748
4000	0.677	0.750	0.912	_	_	0.717	0.904	0.731	0.747	0.965	0.801
5000	0.724	0.817	0.942	_	_	0.778	0.957	0.930	0.773	0.965	0.861
6000	0.729	0.806	0.956	_	_	0.787	0.966	0.947	0.773	0.956	0.865
7000	0.729	0.806	0.956	_	_	0.787	0.966	0.947	0.773	0.956	0.865

Table 5: Systems performance per language when trained on italki.

Experiment #4: Tubasfs system performance on balanced italki. The final experiment, which was to identify the impact of large and balanced dataset on the system's performance, focused on training the system only on those languages (Arabic, Chinese, Japanese, Korea and Spanish) where their training sizes is more than 11,000 documents. We found that the system performance when trained on 7000 documents improved noticeably for Arabic, Chinese and Spanish. Moreover, the average system accuracy also improved.

Training Data	ARA	CHI	JPN	KOR	SPA	AVG
1000	0.608	0.657	0.670	0.703	0.716	0.671
2000	0.657	0.708	0.680	0.635	0.764	0.689
3000	0.701	0.756	0.868	0.729	0.784	0.768
4000	0.722	0.778	0.908	0.737	0.806	0.790
5000	0.774	0.845	0.966	0.943	0.831	0.872
6000	0.753	0.833	0.966	0.952	0.834	0.868
7000	0.750	0.839	0.966	0.952	0.840	0.869

Table 6: Systems performance per language when trained on italki.

The previous experiments have indicated that the availability of a large, and balanced, corpus could significantly improve NLI system performance. From those experiments we identify the need to increase the training size for some of the languages in italki[6]

Experiment #5: Testing for generalization. The previous experiment (experiment #1) showed that the performance of NLI systems when they are trained on italki may potentially be much lower than when they are trained on TOEFL dataset. One of the reasons could be because of the differences between italki and TOEFL from a number of aspect (see section 3.2 for more details). However, the main reason, as highlighted through experiments #2, #3 and #4, is due to the unbalanced size of the training set for some languages in italki.

Since one of the main goal of generating a classification model on a training set for any problem is to generalize the classification model to unseen data. In this experiment, we evaluate the appropriateness of italki for the task of NLI in terms of model generalization. Machine learning algorithms are trained and tuned on a set of training data in the hope that they can perform well on real world data, i.e., generalize to unseen data. One of the most suitable evaluation of the corpus to achieve this goal is via transfer learning.

We trained the selected NLI systems in section 4 on one corpus and tested them on another corpus to examine how well they generalize to data in other corpora (unseen data). We note that in this experiment, we use the unbalanced version of italki, i.e., we use the training set available for all the languages (see Table 1 for more detail).

Table 7 shows the result of the systems evaluation when trained on one corpus (such as italki) and tested on another corpus (such as TOEFL). The systems generalize better when training them on italki than on TOEFL. The empirical results indicate that italki is a better corpus for model generalization.

Table 8 shows the accuracy gain by individual system when trained on italki and tested on TOEFL. Uvic-NLP generalizes the most to unseen data while NLI-ISU generalizes the least. Groningen is second in place followed by Tubasfs in third place.

5 Conclusions

Native Language Identification (NLI) has benefited from the availability of data and advances in machine learning algorithms. Supervised approaches, where machine learning algorithms are trained on labelled data, to classifying the native language of author of text is the dominant approach

[6]Due to time constraint we have reported empirical result for a specific size of training set.

			Test	
			TOEFL	italki
Train	TOEFL	groningen	_	0.4042
		tubasfs	_	0.4035
		NLI-ISU	_	0.3374
		Uvic-NLP	_	0.4136
	italki	groningen	0.5879	_
		tubasfs	0.5807	_
		NLI-ISU	0.5035	_
		Uvic-NLP	0.6177	_

Table 7: Inter-corpus Performance based on F1 measure.

Systems	Generalization difference
Uvic-NLP	0.2041
Groningen	0.1837
Tubasfs	0.1777
NLI-ISU	0.1666

Table 8: Systems generalization measure.

to NLI problem. Although there are a number of corpora available for the NLI task, there exist some limitations in using them to train machine learning classifiers. In this study, we presented a web-scraped corpus (italki) which is larger than the current corpora. We have evaluated several publicly available systems, which performed well in the NLI 2017 shared task, on italki. We have empirically demonstrated that the current approaches, mainly shallow learning where the selected NLI systems utilize, benefit from a training data many times larger than the training data of the TOEFL corpus (which is the most heavily used corpus in previous work). We have evaluated the proposed corpus to identify its contribution to system's generalization to unseen data. We found that systems trained on italki generalize better than those trained on existing corpora.

From our experiment we have identified some limitations in italki. Chiefly, the data imbalance, where for a few languages (Telugu, Hindi and German) the training data is vastly smaller than for other languages. We aim to explore two approaches to address this issue: (i) to collect more data for those languages with small data size, and (ii) to explore the possibility of text generation model (which are based on training deep learning algorithms on the current data set) for automatically generating text for some of the languages with small data size.

Because of the unavailability of large training data, deep learning algorithms, unlike in other natural languages processing tasks, have not made headway in NLI task. Italki provides large quantities of training data. It allows us to experiment with deep learning classifiers and evaluate their performance on NLI. This narrowing of the gap between deep and shallow learning should serve as a motivation for further application of deep learning to NLI, which we aim to investigate in the future.

6 Acknowledgement

This work was supported by the CRITiCaL - combating cRiminals In The CLoud project (EPSRC ref: EP//M020576//1).

References

Brooke, J. & Hirst, G. (2012), Robust, Lexicalized Native Language Identification, *in* 'COLING2012: Conference on Computational Linguistics', The COLING 2012 Organizing Committee, Mumbai, India, pp. 391–408.

Chan, S., Jahromi, M. H., Benetti, B., Lakhani, A. & Fyshe, A. (2017), Ensemble Methods for Native Language Identification, *in* 'BEA2017: Workshop on Innovative Use of NLP for Building Educational Applications', Association for Computational Linguistics, Copenhagen, pp. 217–223.

Estival, D., Gaustad, T., Pham, S. B., Radford, W. & Hutchinson, B. (2007), Author profiling for English emails, *in* 'PACLING2007: Conference of the Pacific Association for Computational Linguistics', Melbourne, Australia, pp. 263–272.

Gibbons, J. (2003), *Forensic Linguistics: An Introduction to Language in the Justice System*, John Wiley & Sons.

Granger, S., Dagneaux, E., Meunier, F. & Paquot, M. (2002), *International corpus of learner English*, Presses universitaires de Louvain.

Jarvis, S. & Crossley, S. A. (2012), *Approaching Language Transfer Through Text Classification: Explorations in the Detection based Approach*, Vol. 64, Multilingual Matters, Bristol, UK.

Koppel, M., Schler, J. & Zigdon, K. (2005), 'Automatically determining an anonymous author's native language', *Intelligence and Security Informatics* pp. 41–76.

Kulmizev, A., Blankers, B., Bjerva, J., Nissim, M., Van Noord, G., Plank, B. & Wieling, M. (2017), The Power of Character N-grams in Native Language Identification, *in* 'BEA2017: Workshop on Innovative Use of NLP for Building Educational Applications', Association for Computational Linguistics, Copenhagen, pp. 382–389.

Malmasi, S., Evanini, K., Cahill, A., Tetreault, J., Pugh, R., Hamill, C., Napolitano, D. & Qian, Y. (2017), A Report on the 2017 Native Language Identification Shared Task, *in* 'BEA2017: Workshop on Innovative Use of NLP for Building Educational Applications', Association for Computational Linguistics, Copenhagen, pp. 62–75.

Nigam, K., Lafferty, J. & Mccallum, A. (1999), Using Maximum Entropy for Text Classification, *in* 'IJCAI1999: Workshop on Machine Learning for Information Filtering', Stockholm, Sweden, pp. 61–67.

Rama, T. & Coltekin, C. (2017), Fewer features perform well at Native Language Identification task, *in* 'BEA2017: Workshop on Innovative Use of NLP for Building Educational Applications', Association for Computational Linguistics, Copenhagen, pp. 255–260.

Rozovskaya, A. & Roth, D. (2011), Algorithm Selection and Model Adaptation for ESL Correction Tasks, *in* 'ACL2011: Meeting of the Association for Computational Linguistics)', Portland, Oregon, USA.

Tetreault, J., Blanchard, D. & Cahill, A. (2013), A report on the first native language identification shared task, *in* 'BEA2013: Workshop on innovative use of NLP for building educational applications', Association for Computational Linguistics, Atlanta, Georgia, pp. 48–57.

Tetreault, J., Blanchard, D., Cahill, A. & Chodorow, M. (2012), Native Tongues, Lost and Found: Resources and Empirical Evaluations in Native Language Identification, *in* 'COLING2012: Conference on Computational Linguistics', Vol. 2, Mumbai, India, pp. 2585–2602.

Tofighi, P., Köse, C. & and Leila Rouka (2012), 'Author's native language identification from web-based texts', *International Journal of Computer and Communication Engineering* **1**(1), 47–50.

Vajjala, S. & Banerjee, S. (2017), A study of N-gram and Embedding Representations for Native Language Identification, *in* 'BEA2017: Workshop on Innovative Use of NLP for Building Educational Applications', Association for Computational Linguistics, Copenhagen, pp. 240–248.

Yang, Y. & Liu, X. (1999), A re-examination of text categorization methods, *in* 'Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 42–49.

Reflexives in Universal Dependencies

Sonja Marković, Daniel Zeman
Charles University, Faculty of Mathematics and Physics, Prague, Czechia
{markovic,zeman}@ufal.mff.cuni.cz

Abstract
We explore the annotation of reflexives in Universal Dependencies (UD) 2.2 treebanks (Nivre et al., 2018), with a stronger focus on Slavic languages. We have tried to find out if the current guidelines are transparent and clear enough for the annotators to follow them successfully. We point out a number of inconsistencies in the current annotation across languages, and propose improvements—sometimes of the guidelines, but mostly of the annotation. The goal of the paper is to contribute to more consistent annotation of reflexives in future releases of UD, which, in turn, will enable broader cross-linguistic studies of this phenomenon.

Keywords: Universal Dependencies, reflexive pronoun, reflexive construction, annotation consistency.

1 Introduction

Reflexive verbs can be found in a significant number of languages, varying from language to language (Geniušienė, 1987, p. 361). This term can be used for verbs with a reflexive marker **(RM)** as their element (affix, inflection, etc.) or as a part of their environment (particle, pronoun, etc.) (Geniušienė, 1987, p. 237). The term 'reflexive' implies that the main function of the reflexive marker is to mark reflexivity, i.e., that a reflexive marker is coreferential with the subject of the clause in which it appears. Nonetheless, marking reflexivity is not the only function of the reflexive marker; it is just one of many functions (Svoboda, 2014, p. 1). The term 'reflexive' is certainly ambiguous but it is broadly used in literature and we will use it in this paper, too. Reflexive markers can be (Geniušienė, 1987, p. 242 and 303):

- affixal morphemes (e.g. prefixes like *t-* in Amharic; suffixes like *-l-* and *-n-* in Turkic languages or *-sja* in Russian; infixes like *-mi-* in Lingala);
- changes in the verbal paradigm (e.g. a change in the agreement paradigm, or a special reflexive conjugation);
- a word or phrase (refl. pronoun like *zibun* "oneself" in Japanese, or a series of pronouns like the Swedish *mig/dig/oss/er/sig*);
- a more or less desemanticized noun meaning "soul", "head", "body", "self" etc., sometimes with a possessive (Basque *buru* "head", e.g., *nere burua* (lit. *my head*) "myself", *bere burua* (lit. *his head*) "himself").

The feature of reflexivity can also apply to modifiers that are neither reflexive verbs nor their arguments. This is the case with reflexive possessives, which indicate that the modified noun is possessed by, or relates to the subject. The Czech example in Figure 1 shows both a reflexive pronoun *(sebe)* and a reflexive possessive *(svého)*.

Both words are coreferential with the subject of the clause, here *Jana*. Thus the two people registered were Jana and Jana's brother (not someone else's brother). The reflexive pronoun is a noun phrase of its own, while reflexive possessives are typically embedded in larger noun phrases. In this paper we focus on various functions of reflexive markers that replace noun phrases or appear at positions similar to those of noun phrases; possessives are not relevant for us, and we will mostly ignore them from now on.

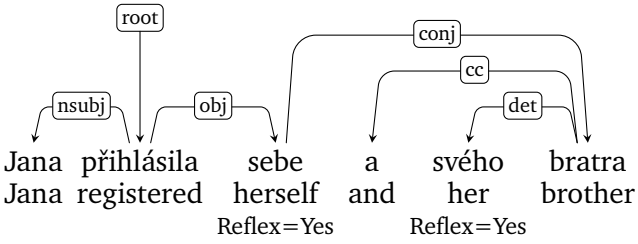


Figure 1: Reflexive object and possessive.

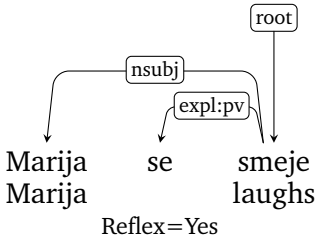


Figure 2: Inherently reflexive.

Furthermore, reflexives in some languages have the same or similar form as intensifiers (emphatic pronouns) such as *himself* in

(1) *John **himself** built this house.*

While in most European languages intensifiers differ from reflexive pronouns (e.g. German *selbst:sich*; Italian *stesso:se*), in many other languages (Turkic, Finno-Ugric, Indic, Persian...), intensifiers are identical with reflexives in form, although not in distribution (König and Siemund, 2000, p. 41). Due to the limited space (and to the language groups focused on in this paper), we do not take intensifiers into account.

Although reflexivity has been intensively studied for the last couple of decades, both in individual languages and from a cross-linguistic perspective, annotating all the functions of all reflexive markers is a complex and sensitive task (Kettnerová and Lopatková, 2014).

In the present paper, we look at reflexives in the context of one particular treebank annotation scheme—Universal Dependencies **(UD)** (Nivre et al., 2016). The nature of the Universal Dependencies project makes annotating reflexive markers particularly difficult, since it aims at developing cross-linguistically consistent treebank annotation for many languages. Discussing the annotation options for reflexives in their various functions is thus very important. This paper is just a starting point for a more comprehensive study of reflexives in UD. At present, such a study is complicated by various imperfections in the data, which we emphasize below. We hope to contribute to better annotation of RMs in future releases of UD; in particular, we have the following goals:

- to present an overview of syntactic and semantic functions of RMs in Slavic languages and to examine their current and desired annotation;
- to present a brief review of selected RMs in Romance and Germanic languages;
- to propose improvements in order to make the data more consistent;
- possibly also to suggest how the guidelines could be made more transparent.

We do not discuss the appropriateness of classifying the RM as a pronoun (in languages where it has a form of a reflexive clitic). In our opinion, this is a controversial question, but we are not sure that a more widely acceptable solution exists.

2 Detecting Reflexives in the UD Treebank Data

In order to make any cross-linguistic claims about reflexives, one must be able to recognize them in corpora. In Universal Dependencies, reflexive words should be annotated with the feature Reflex=Yes. Hence, the feature is our primary source of information; but it apparently has not been used everywhere it should. In UD 2.2, 56 treebanks use the feature.[1] Out of the 71 languages covered in UD 2.2, the Reflex

[1] http://universaldependencies.org/ext-feat-index.html#reflex

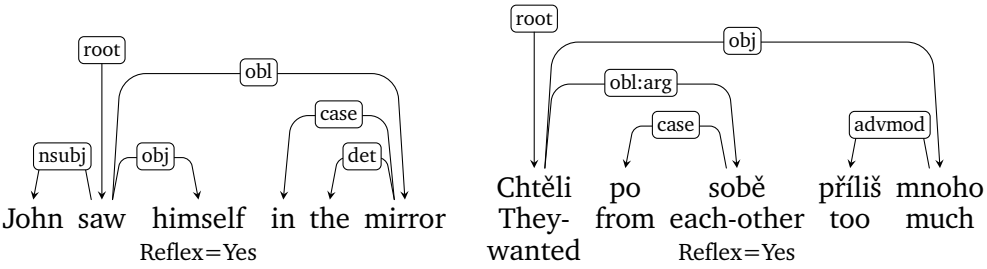


Figure 3: True reflexive (left) and reciprocally used reflexive (right).

feature is present in 44 languages from various language families (Indo-European, Afro-Asiatic, Uralic, Turkic, Mongolic, Dravidian).

In the treebanks where Reflex=Yes is found, it mostly occurs with the PRON part of speech (pronoun) or with DET (determiner; used for reflexive possessives in some languages). A few treebanks use the feature with verbs, adjectives or particles.

There are languages that have reflexive pronouns, yet their treebanks fail to mark them with Reflex=Yes. The list omits, for instance, Hindi, although Hindi has reflexive pronouns such as *apne*. Our observations in the present work are mostly limited to treebanks where the feature is used.

3 Relations / Constructions

3.1 Core and Oblique Dependents

In the simplest case, a reflexive pronoun is used as an object or an oblique dependent of the verb. An irreflexive personal pronoun could occur in the same position, and the syntactic structure is the same regardless of whether the pronoun is or is not reflexive. In the present work, we refer to these cases as *true* or *semantic reflexivity*.

In some languages reflexive pronouns are used also with reciprocal meaning (other languages have dedicated reciprocal pronouns such as "each other" in English). For instance in German

(2) *Braut und Bräutigam haben* ***sich*** *geküsst.* "The bride and groom kissed **each other**."

Here the reflexive pronoun is coreferential with the plural subject as a whole. With regard to semantic roles, each of the two individuals takes the role of the subject and the object at the same time; syntactically however, they function as the subject and the RM as object or oblique dependent (see the second example in Figure 3).

The UD annotation guidelines do not distinguish true reflexives from reciprocally used RMs. They distinguish whether the RM occurs in place of a direct object (labeled

obj), indirect object (iobj) or oblique dependent (labeled obl or one of its subtypes such as obl:arg).

3.2 Inherently Reflexive Verbs

Certain verbs in certain languages require a reflexive clitic without assigning it any semantic role or syntactic function. Traditional grammar treats the clitic as a free lexical morpheme, which is part of the verbal lexeme.

For example, the Slovenian verb *smejati se* "to laugh" is inherently reflexive and never occurs without *se* (see Figure 2). Here the UD guidelines specify that the RM shall be attached as expl:pv (standing for "expletive, subtype pronominal verb").[2]

Some authors restrict the class of inherently reflexive verbs to those that do not have irreflexive counterparts from which they can be derived (Svoboda, 2014, p. 11). Others claim that many transitive verbs have an inherently reflexive variant that is not semantically equivalent to the transitive form, though it is related to it in some way. However, the nature of this semantic relation cannot always be captured easily (Waltereit, 2000, p. 257). Hence it may be difficult to evaluate whether its meaning is "different enough" to rule out the possibility that the reflexive pronoun is just a normal object.

Sometimes a translation into another language may provide a clue. For example, the Latvian *mācīt kam* corresponds to English "to teach somebody", while the reflexive *mācīties* corresponds to "to learn". The difference in translation suggests that the reflexive verb is semantically different and can thus be considered a derived inherently reflexive verb. A more general (but vaguer) clue results from comparing the action performed by the actor of the verb: in Bulgarian *върна нещо някъде (vărna nešto njakăde)* "to return something somewhere", the actor takes an object and moves it in space; even if the object is animate and can move independently, some physical or mental force is applied to make it move in the right direction, quite possibly against its own will. On the other hand, *върна се някъде (vărna se njakăde)* "to return somewhere" describes independent and free movement of the actor to a place where they have been in the past. Again, we can conclude that it describes a different action and is no longer transitive.

The decision is easier if the irreflexive counterpart is intransitive, as in Spanish *ir* "to go" (irreflexive, intransitive) vs. *irse* "to leave" (inherently reflexive).

3.3 Reflexive Passives and Impersonal Constructions

Reflexives in some languages have grammaticalized into markers of alternations in voice (diathesis). They serve as additional means of expressing passive of transitive verbs (especially if their object is inanimate): the original object becomes subject, the

[2] http://universaldependencies.org/u/dep/expl.html#Reflexives

original subject is removed, and a reflexive appears in object position. If a reflexive clause of the form "X did itself" can be paraphrased as "X was done" or "Somebody did X", it is a reflexive passive.[3] The guidelines specify that the RM be attached by the relation `expl:pass` in these cases. For example, in Upper Sorbian

(3) *Serbski institut je so k 1. januarej 1992 wot Swobodneho stata Sakska wutworił.* "The Sorbian Institute was created on January 1, 1992 by the Free State of Saxony."

the RM *so* forms a reflexive passive.

Although the construction is presented as a variant of the passive voice, the verb is actually in its active form; as (Sussex and Cubberley, 2011, p. 448) note, the verb-reflexive complex "is not unlike the Greek middle voice in function and meaning."

Another category, representing another shade of meaning, is called in the literature anticausative (Svoboda, 2014, p. 1–2), decausative or inchoative (Silveira, 2016, p. 116). Examples include Czech and Portuguese

(4) *Dveře se otevřely.* "The door opened."

(5) *O vaso se quebrou.* "The vase broke."

There must be an external cause of the event but the cause is unknown or unidentifiable, and the event seems to come about spontaneously. The difference between anticausative, middle voice and reflexive passive is semantic rather than syntactic, very subtle and hard to discern. Hence it does not seem to be something that can or should be distinguished in UD; we will call them all 'reflexive passive'.

Reflexives can also be used in impersonal constructions, i.e., clauses without subject. They resemble very much the reflexive passive except that the verb is not transitive and there is no object that could be promoted to the subject position. Consequently, the default agreement is triggered on the verb. What exactly it means is language-dependent; for example, in Slavic languages it is the third person, singular, neuter—cf. Polish

(6) *Po Edenie chodziło się nago.* (lit. *Along Eden walked itself nude.*) "One would walk nude in Eden." (Patejuk and Przepiórkowski, 2015).

[3] While the term reflexive passive is used in some classical grammars, other authors regard it as controversial, pointing out differences between reflexive and periphrastic passivization. The stance depends on what one considers the defining properties of passive. In UD, the primary purpose of the `:pass` relation subtype is to signal non-canonical mapping between syntactic relations and semantic roles: auxiliaries in periphrastic passives use `aux:pass`, reflexives use `expl:pass`.

	Sing	Plur
1(|Reflex)	*mir, mich*	*uns*
2(|Reflex)	*dir, dich*	*euch*
3|Masc	*ihm, ihn*	*ihnen, sie*
3|Fem	*ihr, sie*	*ihnen, sie*
3|Neut	*ihm, es*	*ihnen, sie*
3|Reflex	*sich*	*sich*

Table 1: German object and reflexive pronouns. The two forms are dative and accusative; some pronouns have one form for both cases.

The reflexive marker should be attached as `expl:impers` in these cases.

Silveira (2016, p. 124) notes that in some languages impersonal construction can contain even a transitive verb. The 'internal argument' (object) then does not become subject (unlike in passive). It keeps the accusative case and does not trigger verb agreement in person and number; the verb stays in the default third person singular: Spanish

(7) *Se observa cambios en la economia.* "Changes are observed in the economy."

3.4 Double Function / Haplology

A reflexive marker can have a double function. For instance, if an inherently reflexive verb is used in an impersonal construction, there will be just one RM, not two (Patejuk and Przepiórkowski, 2015). Another point is that sometimes one reflexive is shared by two verbs, as in Slovenian, with one inherently reflexive verb controlling another:

(8) *Bal se je smejati.* "He was afraid to laugh."

The guidelines currently do not specify the preferred solution of such cases.

4 Reflexives in Germanic Languages

While English has two parallel sets of irreflexive and reflexive pronouns, in many other languages only the third person has a special reflexive form. First and second person pronouns have the same form whether they are used reflexively or not, but the reflexive usage can be easily recognized because the verb form and the subject pronoun is in the same person and number as the object pronoun. Table 1 shows the pronouns in German.

Table 2 shows that 13 out of 19 Germanic treebanks use the Reflex feature. The remaining 6 treebanks indeed contain reflexive pronouns but they do not tag them as such. Only 6 treebanks use the feature also with 1st and 2nd person pronouns,

Treebank	R12	R3	obj	iobj	obl	expl:pv
Afrikaans	4	3	41		14	
Danish		34	77	12	8	
Dutch Alpino		26	22	2	5	69
Dutch LassySmall		22	28	0	3	69
English EWT	3	2	52	8	15	
English GUM	3	3	54	10	23	
English LinES	4	17	49	2	23	
English PUD		5	80		20	
Faroese		18	67		28	
German GSD	3	55	73	8	3	12
Gothic	1	37	18	9	36	
Norwegian Bokmaal		40	69	14	13	
Norwegian NynorskLIA		16	77	5	14	

Table 2: **R12** = Number of first and second person non-possessive reflexives per 10,000 tokens. **R3** = Number of third person non-possessive reflexives per 10,000 tokens. **obj** = Percentage of reflexives attached as objects. **iobj** = % indirect objects. **obl** = % oblique arguments and adjuncts. **expl:pv** = % inherently reflexive verbs.

although these pronouns are sometimes used reflexively in the other treebanks, too.

Most RMs are attached as true reflexive arguments (`obj`, `iobj`, `obl`). Only in German and Dutch we see some inherently reflexive verbs. They should probably appear elsewhere too, with the exception of English. For instance, we believe that Norwegian *Jeg føler meg som…* "I feel (myself) like…" should be inherently reflexive.

5 Reflexives in Romance Languages

The situation is even less satisfactory in the Romance languages (Table 3). Only 7 treebanks out of 21 tag their RMs with Reflex=Yes. In addition, the Italian ISDT treebank does not use the feature but it uses the reflexive-specific relations `expl:pass` and `expl:impers`. Furthermore, two French treebanks use Reflex=Yes for disjoint sets of pronouns: Sequoia for the clitics *se, me, te, nous, vous,* GSD for the tonic reflexives such as *lui-même, eux-mêmes*.

Some treebanks label all reflexive clitics as core arguments: Spanish GSD as indirect objects, Italian ParTUT (and according to Silveira (2016, p. 126, 129) also French GSD) as direct objects. It is obvious that a significant number of instances are non-argumental and should thus be labeled as expletives.

The Sequoia treebank of French uses just the universal label `expl` for all non-argumental RMs and does not further distinguish inherent, passive and impersonal constructions. This solution is advocated by Silveira (2016, p. 143) who says that the distinction is a source of uncertainty for both parsers and annotators.

Treebank	R12	R3	obj	iobj	obl	expl	:pv	:pass	:imp
French GSD	0	2	2		50				
French Sequoia	5	43	9	3		88			
Italian PUD		1			67				
Italian ParTUT		3	100						
Romanian Nonstd		30	0	2	2		71	15	4
Romanian RRT		180	2	2	0	0	53	28	3
Spanish GSD	2	123	0	99	0				

Table 3: Romance treebanks. **R12**, **R3**, **obj**, **iobj**, **obl** and **expl:pv** – see Table 2. **:pass** = % passive RMs. **:imp** = % impersonal RMs.

Romanian differs from the other Romance languages by having two sets of reflexive pronouns, one in the accusative, and one in the dative (Cojocaru, 2003). The Romanian treebanks do distinguish various subtypes of `expl` as defined by the guidelines and summarized in Sections 3.2 and 3.3. Reflexives analyzed as core arguments are much less frequent than expletives. There is also a Romanian-specific relation, `expl:poss`, used for possessive or benefactive meaning of dative clitics (note that this construction is different from the reflexive possessives mentioned in Section 1).

6 Reflexives in Slavic Languages

Slavic personal pronouns, including reflexive pronouns, have full and clitic forms. Unlike Germanic and Romance languages, the same form is used in all persons and numbers. The full forms occur in all cases except nominative and vocative, and are used to encode true reflexivity, i.e., a nominal that is coreferential with the subject of the clause. The clitic forms are used as true reflexives, too, but they often have other grammatical functions listed in Section 3. There are one or two reflexive clitics per language, which can be characterized as accusative *(se/sa/so/się/sę)* and dative *(si/sej)* forms. In East Slavic (and also in Baltic) languages the reflexive clitic has become a suffix of the verb (Geniušienė, 1987, p. 241).

Clitics and full pronouns are semantically equivalent when they function as true reflexives, and the choice of one of them over the other is made for prosodic and/or pragmatic reasons (topic-focus articulation). The default form for expressing true reflexivity in South and West Slavic languages is the clitic form, while the full form is reserved for expressing emphasis or contrast, when the reflexive pronoun is coordinated with another noun phrase, or after prepositions. The full form can only be used in truly reflexive and reciprocal contexts (Svoboda, 2014, p. 5–6).

6.1 Annotation in Slavic Treebanks

Table 4 gives a summary of RMs in West and South Slavic treebanks. Fourteen treebanks (out of 15) use the Reflex=Yes feature. The first observation is that RMs

Treebank	RM	obj	iobj	obl	expl	:pv	:pass	:imp
Bulgarian	205	0	0	0	98			
Croatian	146	16	0	0		75		
Czech CAC	183	1	1	7		67	23	
Czech CLTT	133	0		1		24	74	
Czech FicTree	366	5	3	10		75	6	
Czech PDT	171	5	2	6		67	19	
Czech PUD	190	9	2	5		68	13	
Old Church Slavonic	28	13	8	77				
Polish LFG	272	2	4	4		85		4
Polish SZ	225	2	2	4		91		
Serbian	140	0		1	98			
Slovak	290	1	1	4		85	7	
Slovenian SSJ	172	1	0	3	95			
Upper Sorbian	178	8		1		40	50	

Table 4: Slavic treebanks. **RM** = Number of non-possessive reflexives per 10,000 tokens. Other columns as in Table 2; `compound` instead of `expl` in Serbian.

are much more frequent than in the Germanic and Romance languages. The counts include both clitics and full pronouns, but the latter are only a small fraction of the whole. Two thirds or more are the occurrences of the clitic *se*.

We do not include East Slavic treebanks in the overview because reflexive clitics are not independent words there. The current UD guidelines actually assume that reflexive verbs should be treated as multi-word tokens, consisting of the verb proper and the clitic suffix. For example, Russian *проснуться (prosnut'sja)* "to wake up" would be split to two syntactic words, *prosnut'* and *sja*, which would make it parallel to the other Slavic languages (e.g. Czech *vzbudit se* "to wake up"). However, this is not what we find in the data. The reflexive verbs are kept together and marked either with Voice=Mid (in Russian and Belarusian) or nothing at all (in Ukrainian). Note that the corresponding verb in Spanish is also sometimes[4] written together with the clitic (*despertarse* "to wake up") but in UD it is split into two words *(despertar se)*. East Slavic languages express true reflexivity using the full pronoun, *sebja* (Svoboda, 2014, p. 29). However, although *-sja* no longer functions as a truly reflexive marker, it can mark the other functions which we investigate in this study.

The next observation is that Old Church Slavonic (OCS) is an outlier. Only true reflexives are tagged as pronouns with Reflex=Yes. The vast majority of occurrences of *sę* are tagged `AUX` and attached via the dependency relation `aux`. This is perhaps inherited from the original PROIEL annotation, but it does not seem to be in accord with the UD guidelines. Supposedly, these occurrences of *sę* function as the RMs for

[4] In the infinitive and imperative.

inherent reflexives, passives and impersonal constructions.

As for dependency relations, the non-AUX RMs in OCS are attached as obj, iobj or obl, which supports the hypothesis that only true reflexives are annotated in the way described in Section 3.

Three other languages (Bulgarian, Serbian and Slovenian) make no difference between the various functions of the RM *se*. They uniformly attach it as expl (in case of Bulgarian and Slovenian) or compound (Serbian). Samardžić et al. (2017, p. 41, 42) explain the reasoning behind their use of compound for all instances of *se*. They view this form as a detachable morpheme belonging to the verb to which it is attached both in lexical and morphological sense. In their view, *se* is not just a prosodic variant of the full reflexive pronoun. In fact, they claim that it is not a pronoun at all, and consequently, it should be analyzed in the same way in all its uses: as a free morpheme marking absence of one of the verb's core arguments. They also state that the different functions noticeable in the other treebanks are higher-level interpretations of the same syntactic form, which should not be part of UD. While we can agree that in many cases this is true, there are still cases where *se* is a true reflexive pronoun, that is, both full and clitic form are possible and it is commutable with clitics of irreflexive personal pronouns. Furthermore, if a single relation is used for all functions other than true reflexives and reciprocals, expl seems to be a better solution than compound, as argued by Silveira (2016) and codified by the UD guidelines.

Figures 4 and 5 show the transitive verb *smatrati* "to consider" in Serbian and Croatian, in both cases in a passive construction. The "compound everywhere" approach is not able to distinguish this from an inherently reflexive verb. Figure 6 is an example of *se* functioning as a core object. We believe that the corresponding Serbian sentence should use obj here too (instead of compound). Therefore, the annotation should differentiate between true reflexivity and the other functions *se* can have.

Polish distinguishes expl:impers but not expl:pass. On the other hand, Czech, Slovak, Upper Sorbian and Croatian distinguish the passive RMs (Croatian labels them aux:pass instead of expl:pass), but they do not use expl:impers. We would argue that both types of constructions exist in all these languages. For example, the Czech FicTree treebank contains impersonal *zapomnělo se na ně* "they were forgotten". In Polish LFG, the anticausative *Drzwi zamknęły się* "The door closed" could be analyzed as expl:pass but has expl:pv instead. However, genuinely passive examples in Polish seem to be rare, presumably because Polish favors the impersonal constructions that keep objects in the accusative: *Maluje się ściany* "The walls are painted."

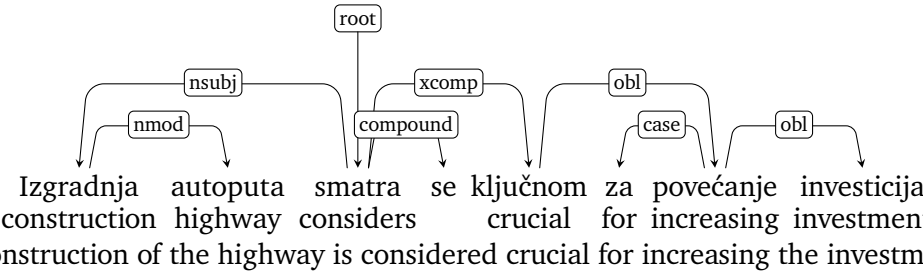


Figure 4: The Serbian treebank always attaches *se* as `compound`. Here, its real function is the reflexive passive.

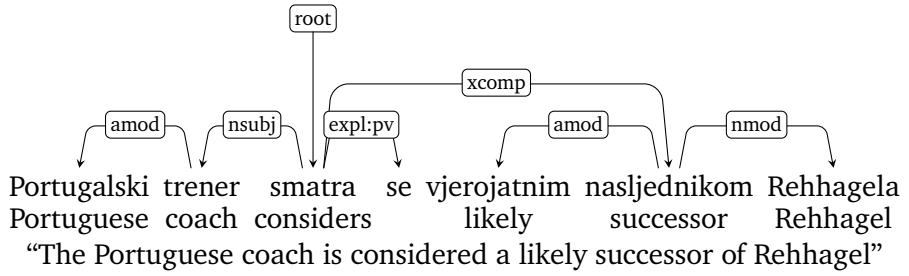


Figure 5: In the Croatian treebank, this *se* is annotated with `expl:pv`, although arguably it has the passive meaning.

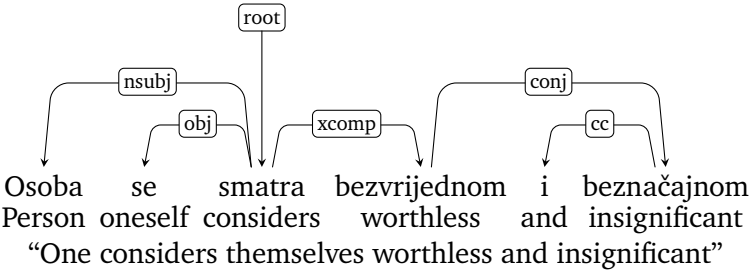


Figure 6: This example is from Croatian but its Serbian counterpart would be similar. Here, *se* is commutable with the full reflexive *sebe*. Hence *se* should be `obj` in this context, in both languages. (We had to shorten the tree due to lack of space, but the full sentence rules out potential ambiguity with `expl:pass`, as *osoba* is the actor: *Osoba ne nalazi nikakvo uporište u sebi samoj, ne vjeruje u sebe, dapače, smatra se bezvrijednom i beznačajnom*. “The person does not find any strength in herself, does not believe in herself, in fact, considers herself worthless and insignificant.”)

7 How to Improve the Treebanks

Annotating all RMs with Reflex=Yes should be easy to achieve because the feature is often tied to just a few lemmas (but it is still helpful for users who do not know the lemmas). Germanic and Romance languages should consider disambiguating reflexive usages of 1st and 2nd person pronouns, as it is already done in German.

As a minimum, all treebanks should distinguish between true reflexivity (`obj` / `iobj` / `obl`) and the non-argumental RMs (`expl` and subtypes). The distinction can often be based on lists of verb lemmas, but it still means that significant manual effort is needed in Bulgarian, Serbian and Slovenian, as well as in some Germanic and Romance languages.

Distinguishing various subtypes of `expl` is optional in UD, yet we would like to advocate at least the distinction between `expl:pv` as a lexical morpheme on one side, and `expl:impers` or `expl:pass` as grammatical means on the other side. Clearer instructions for identifying inherently reflexive verbs are needed, and we have proposed some heuristics in Section 3. The UD guidelines must be modified if East Slavic languages shall keep their reflexive verbs as single syntactic words. The middle voice should then be used in all three languages (it is currently not used in Ukrainian) in order to live up to the UD motto that "same thing should be annotated same way."

Croatian `aux:pass` should be replaced by `expl:pass` and the current `aux` instances should be fixed manually. In Old Church Slavonic, the `AUX/aux` annotation should be replaced by labels that follow the guidelines.

The UD guidelines should specify the priorities when *se* has a double function or is shared by two verbs. In general, the guidelines should provide a broader overview of reflexives with examples from multiple languages; at present, the relevant rules are scattered under various labels.

8 Conclusion

We have shown that the annotation of reflexives in Universal Dependencies is currently unsatisfactory, inconsistent and unpredictable. There is a range of possible causes. First and foremost, constructions with reflexives are a difficult and sometimes controversial issue in many European languages. Multiple analogies with other parts of grammar are available, but most of them have their downsides too. Maintainers of UD treebanks often disagree in their choice of annotation options. More attention should be paid to reflexives in the guidelines and there should be a section devoted to reflexives and discussing all their functions, with examples from many languages. Finally, the data providers should be encouraged to follow a single interpretation of the guidelines, especially in cases where the current annotation can be fixed by an automated procedure.

Acknowledgments

We would like to thank the anonymous reviewers for useful comments. The research was partially supported by the SVV project number 260 453, the grant 15-10472S of the Czech Science Foundation (GAČR), and FP7-ICT-2009-4-249119 (MŠMT 7E11040).

References

Dana Cojocaru. 2003. *Romanian Grammar*. Slavic and East European Language Research Center (SEELRC), Duke University.

Emma Geniušienė. 1987. *The typology of reflexives*. Mouton de Gruyter, Berlin, Germany.

Václava Kettnerová and Markéta Lopatková. 2014. Reflexive verbs in a valency lexicon: The case of Czech reflexive morphemes. In *Proceedings of the XVI EURALEX International Congress: The User in Focus*, Bolzano/Bozen, Italy.

Ekkehard König and Peter Siemund. 2000. Intensifiers and reflexives. a typological perspective. In Zygmunt Frajzyngier and Traci S. Curl, editors, *Reflexives: Forms and Functions. Volume 1*. John Benjamins Publishing Company, Amsterdam/Philadelphia.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, John Bauer, Sandra Bellato, Kepa Bengoetxea, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain

Kahane, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adéday`ọ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Shoval Sadde, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Veronika Vincze, Lars Wallin, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2018. Universal dependencies 2.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher 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 10th International Conference*

on Language Resources and Evaluation (LREC 2016), pages 1659–1666, Portorož, Slovenia. European Language Resources Association.

Agnieszka Patejuk and Adam Przepiórkowski. 2015. An LFG analysis of the so-called reflexive marker in Polish. In *The Proceedings of the LFG'15 Conference*, pages 270–288, Stanford, CA, USA. CSLI Publications.

Tanja Samardžić, Mirjana Starović, Željko Agić, and Nikola Ljubešić. 2017. Universal Dependencies for Serbian in comparison with Croatian and other Slavic languages. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, pages 39–44, Valencia, Spain. Association for Computational Linguistics.

Natalia G. Silveira. 2016. *Designing syntactic representations for NLP: An empirical investigation*. Stanford University, Stanford, CA, USA.

Roland Sussex and Paul Cubberley. 2011. *The Slavic Languages*. Cambridge Language Surveys. Cambridge University Press.

Ondřej Svoboda. 2014. *Functions of the Czech reflexive marker se/si (research master's thesis)*. Universiteit Leiden, Leiden, Netherlands.

Richard Waltereit. 2000. What it means to deceive yourself: The semantic relation of French reflexive verbs and their corresponding transitive verbs. In Zygmunt Frajzyngier and Traci S. Curl, editors, *Reflexives: Forms and Functions. Volume 1*. John Benjamins Publishing Company, Amsterdam/Philadelphia.

Wikinflection: Massive semi-supervised generation of multilingual inflectional corpus from Wiktionary

Eleni Metheniti and Günter Neumann

DFKI
Stuhlsatzenhausweg 3
66123 Saarbrücken

eleni.metheniti@dfki.de, neumann@dfki.de

ABSTRACT

Wiktionary is an open- and crowd-sourced dictionary which has been an important resource for natural language processing/understanding/generation tasks, but a big portion of the available information, such as inflection, is hard to retrieve and has not been widely utilized. In this paper, we are describing our efforts to generate inflectional paradigms for lemmata of the English Wiktionary, by using both the dynamic links of the XML dump file and the static information of the web version. Our system can generate inflectional paradigms for 225K lemmata, with almost 8,5M forms from 1.708 inflectional templates, for over 150 languages, and after evaluating the generation, 216K lemmata and around 6M forms are of high quality. In addition, we retrieve morphological features, affixes and stem allomorphs for each paradigm and form. The system can produce a structured inflectional corpus from any version of the English Wiktionary XML dump file, and could also be adapted for other language versions. The first version of the source code is currently available online.

KEYWORDS: wiktionary, metadata, inflection, corpus, computational morphology.

1 Introduction

1.1 Motivation

Wiktionary is a multilingual, open-sourced project, part of the Wikimedia foundation, which hosts multilingual dictionaries in many target languages. Every lemma in the Wiktionary is sectioned per source language, and contains pronunciation, etymology, definition, derivatives, translations, semantic and inflectional information (if the entry is complete and such information is available). The free and open access and sourcing of this project has established it as a vastly used resource for natural language processing tasks, especially with the use of the domain's XML dump files. These are the source files which are used to dynamically generate the content of a static HTML page per request from the browser; however, the XML file which generates this page only shows links to other lemma pages and utility pages. Providing the XML files allows for natural language processing experts and enthusiasts to quickly and offline extract lexicographic information for many tasks (semantic, phonological, etc.), however, the structured nature of the data can impede or even prevent the extraction of some information, such as inflectional tables.

In this paper, we are describing our attempt to create a multilingual inflectional corpus, with morphological information of affixes and stem allomorphs. We use both the XML dump file and information pulled from the web version of the English Wiktionary, in order to decode machine-readable information (in this case, the dynamic link to an inflectional template) into a human-readable structured file, divided per lemma and per language template. The goal is to generate the inflectional corpus with as little supervision as possible, in order to ensure reproducibility for other users, and extensibility, so that it would be possible in the future to generate dictionaries with updated information or from different editions of the Wiktionary.

1.2 Why inflection

Inflection is the set of morphological processes that occur in a word, so that the word acquires certain grammatical features which either create syntactic dependencies in a phrase (e.g. agreement between nouns and adjectives) or add to the meaning but not change it (e.g. tense in verbs). Inflectional languages have different choices as to which and how these grammatical features will be expressed, for example, most English nouns have four possible forms (singular number, singular number in possessive form, plural number, plural number in possessive form), while nouns in German have eight possible forms (in four cases and two numbers) which vary depending on the gender and the way the noun is declined. The different forms of a word in inflectional languages may be formed by *affixation* (e.g. plural in English nouns), by changes in the stem of the word which will produce a stem *allomorph* (for example, *reduplication*, e.g. plural in Samoan verbs by duplicating part of the stem), or both (e.g. plural in German nouns by *ablaut* and affixation).

$$\begin{aligned} \text{English: } & house_{[+singular]} \rightarrow house - s_{[+plural]} \\ \text{Samoan: } & savali_{[+singular]} \rightarrow sa - va - vali_{[+plural]} \\ \text{German: } & Haus_{[+singular]} \rightarrow Häus - er_{[+plural]} \end{aligned}$$

Inflection has been an ongoing challenge for natural language processing, because of the different levels of morphological richness of every language, the extensiveness of some inflectional paradigms, the low frequency of some forms, the ambiguity when forms are homonyms but have different grammatical properties, to name a few reasons. However, it could prove useful

to tasks that require identification, lemmatization, semantic relations, text generation or use of low-frequency words, because inflection shows the intrinsic bond between forms of the same lemma, and can identify or provide the form with the correct morphosyntactic properties, in tasks such as machine translation, natural language generation and semantic analysis.

2 Previous Work

As a valuable resource for a multitude of natural language processing tasks, there are many available tools to parse the Wiktionary and extract relevant information; the Wikimedia foundation provides the MediaWiki Action API and client libraries in many programming languages, so that users can parse dump files and access information in machine-readable or human-readable ways (MediaWiki, 2018). Concerning the Wiktionary dump files, a discussion page in the domain states the difficulty of parsing a Wiktionary dump file for all its information, because of the presence of dynamic links (Wikipedia contributors, 2017). Most parsing tools, under the auspice of the MediaWiki project or independently developed, either splice the dump file in individual XML page files for easier access (Roland, 2011), or parse and extract specific information which is explicitly stated in the dump file(s), e.g. translations (Acs et al., 2013) or lexical-semantic information (Zesch et al., 2008).

The potential of using Wiktionary as a source of inflectional information has not been untapped, however. Liebeck and Conrad (2015) have created `IWNLP`, a parser for the German edition of the Wiktionary which, with the Lemmatizer module, can produce a mapping from an inflected form to a lemma. First of all, they analyzed the inflectional templates used to dynamically generate inflections, and have re-implemented them in `C#` from the original `Lua`. Then, the tool uses the dynamic link in the page of a lemma, which points to an inflectional template, to generate the inflections (which are then used for the lemmatization task). The accuracy and quality of `IWNLP` is very high, however, so far they have only implemented templates for German nouns, adjectives and most frequently used templates for verbs, and are only using the German edition of Wiktionary.

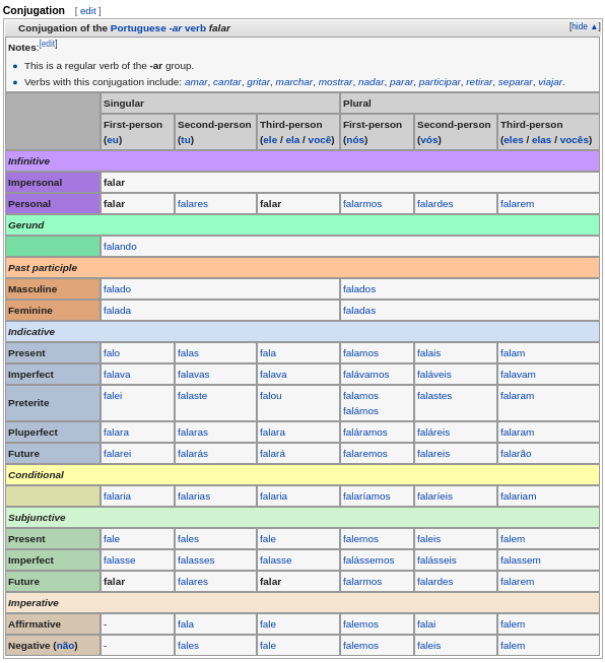
Kirov et al. (2016) followed a radically different approach to gathering inflectional information from the Wiktionary; instead of using the XML dump file, they relied on the static HTML file and used the already generated tables, in order to pull the inflected forms of a lemma. They ensured that their parsing method would yield both the forms and the appropriate features, as noted in the table, and used this information to create a corpus of inflected words with annotated features, using their previously created *UniMorph* annotation schema. Their corpus is of high quality and includes almost 1M entries. However, the practice of pulling information from the web version of Wikimedia pages is highly discouraged, as it could put strain on the servers. In addition, their corpus lacks the information that is included in the dynamic links for a template (presented in detail in Section 3.1) and only includes the lemma and word features.

3 Methodology

3.1 Preliminary work

As mentioned in Section 1.1, a Wiktionary XML dump file contains all the pages available in the Wiktionary website for a specific target language, not in HTML format, but with dynamic links. Whenever there is a request to access a web page, the server runs a script for every component that is encoded, and decodes it into the relevant information. For example, as seen in Figure 1, the web page for the word *'falar'* ('to talk') contains a table with the verb conjugation for the word in Portuguese. This web page is generated by the XML page for *'falar'*

(also included in the dump file in XML format, as seen in Figure 2), which contains links to other word pages (in multiple brackets) and utility pages (in multiple curly brackets). To generate the verb conjugation, this formula is used as a link: `{{pt-conj|fal|ar}}`. Its first parameter refers to the template which should be used, the second parameter is the stem of the word (`fal`) and the third parameter is conjugation information (the suffix `ar`).



Conjugation [edit]

Conjugation of the Portuguese -ar verb *falar* [hide ▲]

Notes [edit]

- This is a regular verb of the **-ar** group.
- Verbs with this conjugation include: *amar*, *cantar*, *gritar*, *marchar*, *mostrar*, *nadar*, *parar*, *participar*, *retirar*, *separar*, *viajar*.

	Singular			Plural		
	First-person (eu)	Second-person (tu)	Third-person (ele / ela / você)	First-person (nós)	Second-person (vós)	Third-person (eles / elas / vocês)
Infinitive						
Impersonal	**falar**					
Personal	**falar**	falares	**falar**	falarmos	falardes	falarem
Gerund						
	falando					
Past participle						
Masculine	falado			falados		
Feminine	falada			faladas		
Indicative						
Present	falo	falas	fala	falamos	falais	falam
Imperfect	falava	falavas	falava	falávamos	faláveis	falavam
Preterite	falei	falaste	falou	falamos falámos	falastes	falaram
Pluperfect	falara	falaras	falara	faláramos	faláreis	falaram
Future	falarei	falarás	falará	falaremos	falareis	falarão
Conditional						
	falaria	falarias	falaria	falaríamos	falaríeis	falariam
Subjunctive						
Present	fale	fales	fale	falemos	faleis	falem
Imperfect	falasse	falasses	falasse	falássemos	falásseis	falassem
Future	**falar**	falares	**falar**	falarmos	falardes	falarem
Imperative						
Affirmative	-	fala	fale	falemos	falai	falem
Negative (não)	-	fales	fale	falemos	faleis	falem

Figure 1: Web page (excerpt) of the lemma *'falar'* in the English Wiktionary.

```
==Portuguese==

===Alternative forms===
* {{l|pt|fallar}} {{qualifier|obsolete}}
* {{l|pt|falá}} {{qualifier|apocopic or eye dialect}}

===Etymology===
From {{etyl|roa-opt|pt}} {{m|roa-opt|falar}}, from {{etyl|la|pt}} {{m|la|fābulārī}}, present
infinitive of {{m|la|fābulor||chat, converse}}.

===Pronunciation===
* {{a|PT}} {{IPA|/fɐˈlaɾ/|lang=pt}}
* {{a|BR}} {{IPA|/faˈla(ɾ)/|lang=pt}}
* {{a|Nordestino}} {{IPA|/faˈla(h)/|lang=pt}}
* {{a|Sul}} {{IPA|/faˈlaɹ/|/faˈlaɻ/|lang=pt}}

===Verb===
{{pt-verb|fal|ar}}

# {{lb|pt|intransitive}} to {{l|en|speak}}; to {{l|en|talk}} {{gloss|to say words out loud}}
#: {{ux|pt|Para de '''falar'''.|Stop '''talking'''.|inline=1}}
#: {{ux|pt|'''Fala'''!|'''Talk'''!|inline=1}}
#: {{ux|pt|'''Fale'''!|'''Talk'''!|inline=1}}
# {{lb|pt|by extension}} to {{l|en|communicate}} by any means
#: {{ux|pt|'''Falamo'''-nos por correio.|We '''communicate''' by mail.|inline=1}}
# {{lb|pt|transitive}} to {{l|en|say}} something
#: {{ux|pt|Para de '''falar''' bobagens.|Stop '''talking''' nonsense.|inline=1}}
#: {{ux|pt|'''Fala''' bobagens.|'''Talk''' nonsense.|inline=1}}
# {{indtr|pt|com}} to {{l|en|talk}} {{l|en|to}}
#: {{ux|pt|Estou '''falando''' com você|I'm '''talking''' to you.|inline=1}}
# {{indtr|pt|para}} to {{l|en|tell}} {{gloss|to convey by speech}}
#: {{ux|pt|Vou '''falar''' para você.|I'm going to '''tell''' you.|inline=1}}
# {{indtr|pt|de|sobre}} to {{l|en|talk}} about
# {{indtr|pt|de}} to {{l|en|speak ill of}}
# {{lb|pt|transitive}} to {{l|en|speak}} {{gloss|to be able to communicate in a language}}
#: {{ux|pt|Em Portugal se '''fala''' português.|In Portugal they '''speak''' Portuguese.}}

====Conjugation====
{{pt-conj|fal|ar}}
```

Figure 2: XML page (excerpt) of the lemma *'falar'* in the English Wiktionary.

The formula links the lemma to a utility page, which contains the template `Template:pt-conj` to generate the inflectional paradigm of *'falar'*. However, upon inspecting the web page of the template[1] and the corresponding XML page (Figure 3a), it is observed that the template is also generated and not explicitly written; as declared with a link, it calls for a module page, `Module:pt-conj` which will generate the template, which in turn will generate the inflectional paradigm of the verb. The module is written in `Lua` (an excerpt can be seen in Figure 3b and the full script is online[2]) and uses the other parameters provided in the inflectional formula for the lemma, `fal` and `ar`.


```
  <page>
    <title>Template:pt-conj</title>
    <ns>10</ns>
    <id>1294753</id>
    <revision>
      <id>32142499</id>
      <parentid>13609370</parentid>
      <timestamp>2015-01-28T14:10:20Z</timestamp>
      <contributor>
        <username>Jberkel</username>
        <id>1580588</id>
      </contributor>
      <comment>use module</comment>
      <model>wikitext</model>
      <format>text/x-wiki</format>
      <text xml:space="preserve">&lt;includeonly&gt;{{#invoke:pt-conj|show}}&lt;/
includeonly&gt;&lt;!--

--&gt;&lt;noinclude&gt;{{documentation}}&lt;/noinclude&gt;</text>
      <sha1>3h1y5upf3gm5iwns5dyhnkyxgfqwjos</sha1>
    </revision>
```


(a) XML page for the template.

```
local function verbData(ending)
   local group
   if ending == 'pôr' or ending == 'por' then
      group = 'er'
   elseif ending == 'erir-defective' then
      group = 'ir'
   else
      group, _ = string.gsub(ending, &quot;%d+$&quot;, &quot;&quot;)
      group = string.sub(group, #group-1)
   end
   if group == &quot;&quot; then
         return nil
   end
   local success, m_verb_data = pcall(require, &quot;Module:pt-conj/data/~&quot;..group)
   if success and m_verb_data[ending] then
      return mw.clone(m_verb_data[ending])
   else
      return nil
   end
end

local function applyFuncToTableValues(tbl, func)
   for k,v in pairs(tbl) do
      if type(v) == 'table' then
         applyFuncToTableValues(v, func)
      else
         tbl[k] = func(v)
      end
   end
end
```

(b) Module script in `Lua` (excerpt).

Figure 3: The template and module to generate `pt-conj`.

Our initial attempt at generating inflections was to replicate the process in which a module generates an inflectional paradigm in a web page, but this approach proved to be unsuccessful. The steps followed were: (a) reading the XML dump file and extracting the lemma pages, the template pages and the module pages, (b) saving the module pages in their individual `Lua` script file, (c) finding the inflectional formula(s) in each lemma (i.e. the template(s) and the required parameters), (d) finding the template(s) linked to the lemma, (e) finding the module linked to the template(s), and (f) run the module with the parameters of the inflectional formula(s). Unfortunately, this approach proved to be unsustainable, because the `Lua` scripts need more data than the given; they require the template pages on how to generate an HTML table, some of the scripts require multiple inflectional templates or data dictionaries that are not available in the XML file etc. The source code of a static HTML page mentions in a comment all the templates and modules that were used to generate the dynamic content, however, this information is not in all cases explicitly stated in the XML files. Therefore, it was not possible to produce significant results without querying every lemma page from the web edition for its comments, and it would be impossible to edit every script without supervision.

Our second attempt focused, at first, on templates; they are available online in static HTML pages, and they are generated by the respective modules. As seen in Figure 4, the generated web page for a template is quite comprehensible: the table includes all the forms of the inflectional paradigm with their respective labels, in the same way that they appear in the lemma web page (Figure 5) – the only difference is that, in place of stem, it includes the link `{{{1}}}`. From our previous attempt, we understood that a dynamic link to a template, for example, `{{pl-decl-adj-owy|róż}}` for the lemma *'różowy'* ('pink'), includes as first parameter the template name, as second parameter the stem of the paradigm, etc., therefore it was easy to

[1] `https://en.wiktionary.org/wiki/Template:pt-conj`
[2] `https://en.wiktionary.org/wiki/Module:pt-conj`

understand that `{{{1}}}` refers to the second parameter of the link[3]; when the module is called, the second parameter will replace the placeholder `{{{1}}}` and form the declension.

declension of *{{{1}}}owy* [hide ▲]

case	singular				plural	
	m pers, m anim	*m inan*	*n*	*f*	*m pers*	other
nominative, vocative	{{{1}}}owy		{{{1}}}owe	{{{1}}}owa	{{{1}}}owi	{{{1}}}owe
genitive	{{{1}}}owego			{{{1}}}owej	{{{1}}}owych	
dative	{{{1}}}owemu				{{{1}}}owym	
accusative	{{{1}}}owego	{{{1}}}owy	{{{1}}}owe	{{{1}}}ową	{{{1}}}owych	{{{1}}}owe
instrumental	{{{1}}}owym				{{{1}}}owymi	
locative				{{{1}}}owej	{{{1}}}owych	

Figure 4: Table from template `pl-decl-adj-owy`.

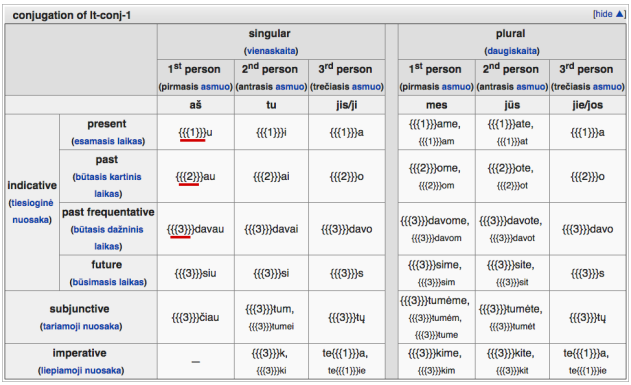
declension of *różowy* [hide ▲]

case	singular				plural	
	m pers, m anim	*m inan*	*n*	*f*	*m pers*	other
nominative, vocative	**różowy**		różowe	różowa	różowi	różowe
genitive	różowego			różowej	różowych	
dative	różowemu				różowym	
accusative	różowego	**różowy**	różowe	różową	różowych	różowe
instrumental	różowym				różowymi	
locative				różowej	różowych	

Figure 5: Table from lemma *różowy*.

The same process applies for templates which require more than two parameters, for example the Lithuanian verb *'gauti'* ('to get') has the dynamic link `{{lt-conj-1|gaun|gav|gau}}` because the template requires four parameters to create the conjugation table (Figures 6, 7) – the last three exist because this inflectional paradigm requires stem allomorphs. This proves to be very interesting, because the template link provides information that sometimes is not mentioned in the entry of a lemma, i.e. the type of inflectional paradigm that the word follows, and the changes that happen to the form beyond affixation.

For our approach, we decided that template information, both dynamic and static, was going to be an integral part of our corpus. In the following section, we will explain how we explored and used it.

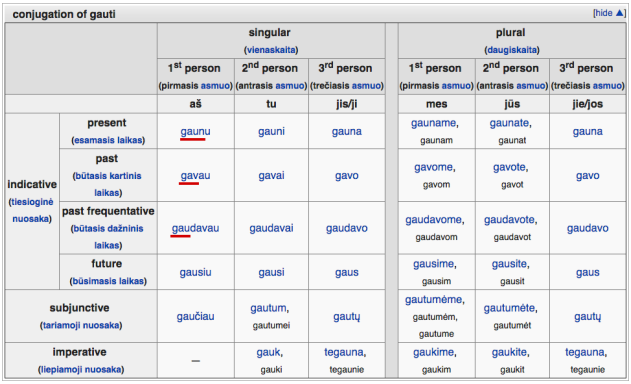


conjugation of lt-conj-1 [hide ▲]

		singular (vienaskaita)			plural (daugiskaita)		
		1st person (pirmasis asmuo)	2nd person (antrasis asmuo)	3rd person (trečiasis asmuo)	1st person (pirmasis asmuo)	2nd person (antrasis asmuo)	3rd person (trečiasis asmuo)
		aš	tu	jis/ji	mes	jūs	jie/jos
indicative (tiesioginė nuosaka)	present (esamasis laikas)	{{{1}}}u	{{{1}}}i	{{{1}}}a	{{{1}}}ame, {{{1}}}am	{{{1}}}ate, {{{1}}}at	{{{1}}}a
	past (būtasis kartinis laikas)	{{{2}}}au	{{{2}}}ai	{{{2}}}o	{{{2}}}ome, {{{2}}}om	{{{2}}}ote, {{{2}}}ot	{{{2}}}o
	past frequentative (būtasis dažninis laikas)	{{{3}}}davau	{{{3}}}davai	{{{3}}}davo	{{{3}}}davome, {{{3}}}davom	{{{3}}}davote, {{{3}}}davot	{{{3}}}davo
	future (būsimasis laikas)	{{{3}}}siu	{{{3}}}si	{{{3}}}s	{{{3}}}sime, {{{3}}}sim	{{{3}}}site, {{{3}}}sit	{{{3}}}s
subjunctive (tariamoji nuosaka)		{{{3}}}čiau	{{{3}}}tum, {{{3}}}tumei	{{{3}}}tų	{{{3}}}tumėme, {{{3}}}tumėm, {{{3}}}tume	{{{3}}}tumėte, {{{3}}}tumėt	{{{3}}}tų
imperative (liepiamoji nuosaka)		–	{{{3}}}k, {{{3}}}ki	te{{{1}}}a, te{{{1}}}ie	{{{3}}}kime, {{{3}}}kim	{{{3}}}kite, {{{3}}}kit	te{{{1}}}a, te{{{1}}}ie

Figure 6: Table from template `lt-conj-1`.

[3] Traditionally, programming languages start indexing elements at zero, therefore the first parameter has position [0], the second [1], and so on.



conjugation of gauti [hide ▲]

		singular (vienaskaita)			plural (daugiskaita)		
		1st person (pirmasis asmuo)	2nd person (antrasis asmuo)	3rd person (trečiasis asmuo)	1st person (pirmasis asmuo)	2nd person (antrasis asmuo)	3rd person (trečiasis asmuo)
		aš	tu	jis/ji	mes	jūs	jie/jos
indicative (tiesioginė nuosaka)	present (esamasis laikas)	gaunu	gauni	gauna	gaunam, gaunam	gaunate, gaunat	gauna
	past (būtasis kartinis laikas)	gavau	gavai	gavo	gavome, gavom	gavote, gavot	gavo
	past frequentative (būtasis dažninis laikas)	gaudavau	gaudavai	gaudavo	gaudavome, gaudavom	gaudavote, gaudavot	gaudavo
	future (būsimasis laikas)	gausiu	gausi	gaus	gausime, gausim	gausite, gausit	gaus
subjunctive (tariamoji nuosaka)		gaučiau	gautum, gautumei	gautų	gautumėme, gautumėm, gautume	gautumėte, gautumėt	gautų
imperative (liepiamoji nuosaka)		–	gauk, gauki	tegauna, tegaunie	gaukime, gaukim	gaukite, gaukit	tegauna, tegaunie

Figure 7: Table from lemma *gauti*.

3.2 Creating inflectional templates and paradigms

In the process of extracting inflectional templates to later use them on lemmata, it is already explained why we could not easily extract them from the XML dump file; thus our other option was to extract them from the respective web pages. While querying the Wiktionary website is not the recommended practice, the number of requests we would have to perform would be significantly lower than the ones performed by Kirov et al., because we would look for templates and not all web pages.

The first step started with the XML dump file; as mentioned, it contains all the pages of the web domain, so we would be able to extract the pages of lemmata and the pages of templates; the goal was to create a dictionary of all the word entries in the Wiktionary which (a) are lemmata and not pages of a form and (b) contain at least one dynamic link to an inflectional template. For example, the words *'falar'*, *'różowy'* and *'gauti'* are added to the dictionary of entries, because they fulfill the needed criteria, but the words *'houses'*[4] or *'architecture'*[5] are not (the former is not a lemma, and the latter does not have any links to inflectional information on Wiktionary). Multiple template links for a single lemma means that the word exists as a lemma in multiple languages, or that it adheres to many inflectional paradigms (e.g. in Figure 8, the lemma *'ring'* in Danish may follow two different noun inflectional paradigms.). In this step, we also create a list of all template names which are used for inflectional paradigms. The template names extracted also had to fulfill some criteria; they needed to contain the words 'noun', 'verb', etc. or the words 'decl', ' conj', etc. in order to not be confused with templates for other linguistic information (e.g. phonology) or utility templates (e.g. 'table' templates which generate the format of an HTML table). In this step, we extracted 454.470 pages of lemmata with inflectional template links (out of the 5.740.594 words in the latest edition of the English Wiktionary dump file) and 7.068 inflectional templates.

We then had to perform a request for the web page of every template; in order to perform these requests only once, we opted to download the HTML files and use the local files to generate the templates. In `Python3`, we used the `BeautifulSoup` library to parse the HTML code and find an HTML inflectional table, and the `pandas` library to convert the table to a data frame. We had to overcome some formatting issues, for example, merged cells which contained labels

[4] `https://en.wiktionary.org/wiki/houses`
[5] `https://en.wiktionary.org/wiki/architecture`

```
'ring': [['da-noun-infl', 'en', 'e'],
         ['da-noun-infl', 'et'],
         ['hu-conj-ok', 'ri', 'n', 'g'],
         ['cs-decl-noun', 'ring', 'ringu', 'ringu', 'ring', 'ringu', 'ringu', 'ringem', 'ringy', 'ringů', 'ringům', 'ringy', 'ringy','rinzích', 'ringy'],
         ['et-decl-riik', 'ring', 'ring', 'i'],
         ['hu-infl-nom', 'ringe', 'e'],
         ['sv-infl-noun-c-ar']]
```

Figure 8: The entry *'ring'* in the word dictionary. The first digits of every template, before the first dash, refer to the language's ISO 639-5 code.

for multiple rows or columns had to be unmerged and duplicated (Ricco, 2017), and some cells contained multiple forms, either phonetic transcriptions in the Latin alphabet or different possible forms – we decided that as much of the available information should be preserved, so duplicate entries had to be made in the inflectional template.

In addition, we decided to convert the extracted morphological features from text to Universal Dependencies morphological feature tags (Nivre et al., 2018); this was done for the sake of uniformity, because authors of different templates had made different choices in the way features were written (e.g. 'singular' vs. 'sing.'). Universal Dependencies were specifically chosen, because they have been used across many NLP applications, from treebanks and annotated data to syntactic parsers, morphological taggers etc. and are extensively documented, therefore our corpus could prove useful for a variety of applications. In order to convert the arbitrary Wiktionary tags to UD tags, we built a database of all the available Universal Dependencies tags, and as many Wiktionary tags and their variations as we could possibly gather. This database however is not exhaustive, as in some cases tags are written abbreviated (e.g. 'masculine' can appear as 'm', 'm.', 'male', 'masc.' etc) or are not in English (e.g. Figure 7).

The greatest challenge, however, came with templates such as `pt-conj` (see Section 3.1); such templates' web pages have no inflectional tables and rely solely on the respective module to generate the inflections. Since we have made the decision not to use any module information, these templates unfortunately could not be parsed, and would not be included in our corpus. Out of the 7.068 saved HTML pages, 2.927 templates had inflectional information in table format that could be parsed. The output of our template reading script produces a dictionary of templates, where every template includes a list of all possible inflected forms, each with their morphological features. Examples of this dictionary's entries can be seen in Figures 9a, 9b.

```
'pl-decl-adj-owy': [['{{{1}}}owy', 'Case=Voc'],
 ['{{{1}}}owe', 'Case=Voc'],
 ['{{{1}}}owa', 'Case=Voc'],
 ['{{{1}}}owi', 'Case=Voc'],
 ['{{{1}}}owe', 'Case=Voc'],
 ['{{{1}}}owego', 'Case=Gen'],
 ['{{{1}}}owej', 'Case=Gen'],
 ['{{{1}}}owych', 'Case=Gen'],
 ['{{{1}}}owemu', 'Case=Dat'],
 ['{{{1}}}owym', 'Case=Dat'],
 ['{{{1}}}owego', 'Case=Acc'],
 ['{{{1}}}owy', 'Case=Acc'],
 ['{{{1}}}owe', 'Case=Acc'],
 ['{{{1}}}ową', 'Case=Acc'],
 ['{{{1}}}owych', 'Case=Acc'],
 ['{{{1}}}owe', 'Case=Acc'],
```

(a) Parsed template: `pl-decl-adj-owy`.

```
{'lt-conj-1': [['{{{1}}}u', 'Mood=Ind'],
  ['{{{1}}}i', 'Mood=Ind'],
  ['{{{1}}}a', 'Mood=Ind'],
  ['{{{1}}}ame', 'Mood=Ind'],
  ['{{{1}}}am', 'Mood=Ind'],
  ['{{{1}}}ate', 'Mood=Ind'],
  ['{{{1}}}at', 'Mood=Ind'],
  ['{{{1}}}a', 'Mood=Ind'],
  ['{{{2}}}au', 'Tense=Past'],
  ['{{{2}}}ai', 'Tense=Past'],
  ['{{{2}}}o', 'Tense=Past'],
  ['{{{2}}}ome', 'Tense=Past'],
  ['{{{2}}}om', 'Tense=Past'],
  ['{{{2}}}ote', 'Tense=Past'],
  ['{{{2}}}ot', 'Tense=Past'],
  ['{{{2}}}o', 'Tense=Past'],
```

(b) Parsed template: `lt-conj-1`.

Figure 9: Entries from the template dictionary.

After the dictionary of templates is made, we used the dictionary of word entries to iterate over every word, and for every template link that the word had, we generated an inflectional

paradigm, as seen in Figures 10, 11. Each form has information for its corresponding inflectional template, morphological data, the part-of-speech tag (depending on the template), the stem used by the template to generate this form, and a list of prefixes, suffixes and infixes if available. The resulting dictionary has 225453 lemmata, which have been matched with 1708 templates in order to generate 8426480 forms, in a total of 199 languages.

```
'różowy': [[['różowy', 'pl-decl-adj-owy', ['Case=Voc'], 'ADJ', ['', 'owy', '']], 'róż'],
  ['różowe', 'pl-decl-adj-owy', ['Case=Voc'], 'ADJ', ['', 'owe', '']], 'róż'],
  ['różowa', 'pl-decl-adj-owy', ['Case=Voc'], 'ADJ', ['', 'owa', '']], 'róż'],
  ['różowi', 'pl-decl-adj-owy', ['Case=Voc'], 'ADJ', ['', 'owi', '']], 'róż'],
  ['różowe', 'pl-decl-adj-owy', ['Case=Voc'], 'ADJ', ['', 'owe', '']], 'róż'],
  ['różowego',   'pl-decl-adj-owy',   ['Case=Gen'],   'ADJ',   ['', 'owego', '']],   'róż'],
  ['różowej', 'pl-decl-adj-owy', ['Case=Gen'], 'ADJ', ['', 'owej', '']], 'róż'],
  ['różowych',   'pl-decl-adj-owy',   ['Case=Gen'],   'ADJ',   ['', 'owych', '']],   'róż'],
  ['różowemu',   'pl-decl-adj-owy',   ['Case=Dat'],   'ADJ',   ['', 'owemu', '']],   'róż'],
  ['różowym', 'pl-decl-adj-owy', ['Case=Dat'], 'ADJ', ['', 'owym', '']], 'róż'],
  ['różowego',   'pl-decl-adj-owy',   ['Case=Acc'],   'ADJ',   ['', 'owego', '']],   'róż'],
  ['różowy', 'pl-decl-adj-owy', ['Case=Acc'], 'ADJ', ['', 'owy', '']], 'róż'],
  ['różowe', 'pl-decl-adj-owy', ['Case=Acc'], 'ADJ', ['', 'owe', '']], 'róż'],
  ['różową', 'pl-decl-adj-owy', ['Case=Acc'], 'ADJ', ['', 'ową', '']], 'róż'],
  ['różowych',   'pl-decl-adj-owy',   ['Case=Acc'],   'ADJ',   ['', 'owych', '']],   'róż'],
  ['różowe', 'pl-decl-adj-owy', ['Case=Acc'], 'ADJ', ['', 'owe', '']], 'róż'],
```

Figure 10: Paradigm (excerpt) for *różowy*.

```
'gauti': [[['gaunu', 'lt-conj-1', ['Mood=Ind'], 'VERB', ['', 'u', '']], 'gaun'],
  ['gauni', 'lt-conj-1', ['Mood=Ind'], 'VERB', ['', 'i', '']], 'gaun'],
  ['gauna', 'lt-conj-1', ['Mood=Ind'], 'VERB', ['', 'a', '']], 'gaun'],
  ['gauname', 'lt-conj-1', ['Mood=Ind'], 'VERB', ['', 'ame', '']], 'gaun'],
  ['gaunam', 'lt-conj-1', ['Mood=Ind'], 'VERB', ['', 'am', '']], 'gaun'],
  ['gaunate', 'lt-conj-1', ['Mood=Ind'], 'VERB', ['', 'ate', '']], 'gaun'],
  ['gaunat', 'lt-conj-1', ['Mood=Ind'], 'VERB', ['', 'at', '']], 'gaun'],
  ['gauna', 'lt-conj-1', ['Mood=Ind'], 'VERB', ['', 'a', '']], 'gaun'],
  ['gavau', 'lt-conj-1', ['Tense=Past'], 'VERB', ['', 'au', '']], 'gav'],
  ['gavai', 'lt-conj-1', ['Tense=Past'], 'VERB', ['', 'ai', '']], 'gav'],
  ['gavo', 'lt-conj-1', ['Tense=Past'], 'VERB', ['', 'o', '']], 'gav'],
  ['gavome', 'lt-conj-1', ['Tense=Past'], 'VERB', ['', 'ome', '']], 'gav'],
  ['gavom', 'lt-conj-1', ['Tense=Past'], 'VERB', ['', 'om', '']], 'gav'],
  ['gavote', 'lt-conj-1', ['Tense=Past'], 'VERB', ['', 'ote', '']], 'gav'],
  ['gavot', 'lt-conj-1', ['Tense=Past'], 'VERB', ['', 'ot', '']], 'gav'],
  ['gavo', 'lt-conj-1', ['Tense=Past'], 'VERB', ['', 'o', '']], 'gav'],
```

Figure 11: Paradigm (excerpt) for *gauti*.

3.3 Evaluating and correcting the paradigms

The generation process was successful, however, it is necessary that we check the quality of the produced paradigms. The first option would be to examine if these words exist in a corpus, and therefore are grammatical formations. However, it would be impossible to find corpora for all 199 languages, and corpora large enough to include forms that are grammatical but might be of very low frequency. We conducted a small-scale experiment to prove that the use of corpora would be neither feasible nor fruitful; we used the inflectional paradigm of the Finnish verb *'taitaa'* and queried some of its forms in *Araneum Finnicum Maius* (Benko, 2016), a Finnish corpus of over 1,2B tokens. forms such as *'taidettaessa'* (second passive infinitive in inessive case) were not found (Figure 12).

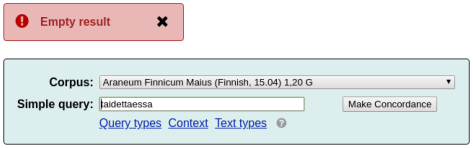


Figure 12: Querying the word '*taidettaessa*' returned no results from *Finnicum Maius*.

Our second option to evaluate our generated inflections would be to look inside the source, the Wiktionary, and check if the generated forms exist in the page of a lemma. This approach

would ensure that our generating process yields the same results as the modules generating templates and lemmata. However, for reasons already stated, we decided to not check every single inflectional paradigm; instead, we created a script which randomly selects one lemma from each template (i.e. 1.708 unique lemmata), finds the web page of the lemma and looks up the presence of all the inflected forms of the lemma. In Table 1 the results of two random evaluations are presented. [6]

	Random evaluation No. 1				Random evaluation No. 2			
Template	***Word***	***All***	***Correct***	***False***	***Word***	***All***	***Correct***	***False***
`la-decl-2nd`	*campus*	12	12	0	*Herostratus*	12	8	4
`de-decl-adj`	*großbürgerlich*	48	48	0	*unmöglich*	48	48	0
`ga-decl-m1`	*gob*	16	12	4	*baneachlach*	16	12	4
`ang-decl-noun-a-n`	*bispell*	8	4	4	*gedal*	8	4	4
`osx-decl-noun-a-n`	*baluwerk*	8	8	0	*god*	8	8	0
`pl-decl-noun-masc-ani`	*palant*	15	15	0	*torbacz*	15	14	1

Table 1: Evaluation results. 'All' refers to the number of forms in the paradigm.

After a few evaluation runs, we first noticed that some templates and template links behaved irregularly compared to others; for example, the lemma *'tocar'* contains the link to the Spanish conjugation template as `{{es-conj-ar|to}}`, but the second parameter does not provide a correct stem to complete the template. Also, as previously seen in Figure 8, the authors of the Danish links did not include stem information in the links, because their templates use the lemma for inflections, but the authors of the Estonian and Hungarian links include the stem and stem allomorphs, and the Czech authors have opted to not generate any inflections with a template, but to input all forms as word allomorphs. We decided to perform a quick revision to templates per language, and create a few exceptions for languages with different dynamic link formatting, because Wiktionary authors of the same language tend to adhere to the same formatting rules. It would be impossible to manually check all the templates, and it would also not be reproducible in case of new or updated templates in Wiktionary.

We also noticed that our evaluation might produce false negatives, because of decoding problems when parsing an HTML page (a problem which appeared in languages such as Serbo-Croatian), or because of irregularities in the inflection of a random word which are caused by extraneous factors (for example, as seen in Table 1, the word *Herostratus* for `la-decl-2nd` produced false negatives because the word is a common noun and does not have a plural form), or because in lemmata with too many generated information the server sometimes fails to execute all `Lua` scripts and generate all content.

Using the Wiktionary as means of evaluation is not optimal, but it is the best available source so far for looking up full inflectional paradigms and words that are grammatical but may not exist in a language corpus. After running the evaluation, our script stores in memory the indices of the false forms per paradigm, and then updates the templates by removing these false forms (or the entire template if it is incorrect) and re-generates the paradigms, which some of them now may be partial but should have good quality forms. The number of generated paradigms and forms may vary per generation, but for random evaluation No. 1, the total number of paradigms was 216.378, generated by 1.537 templates, and yielded 5.970.799 forms, and for random evaluation No. 3, the total number of paradigms was 210.172, generated by 1.521 templates, and yielded 6.024.077 forms. A table for random evaluation No. 3 can be found in Table 2.[6]

[6] Full tables are available online at `https://tinyurl.com/wikinflection`

			Before evaluation and correction		After random eval. No.3 and correction	
Language	*Template*	*Lemmas per temp.*	*Inflections per lemma*	*No. forms*	*Inflections per lemma*	*No. forms*
Finnish	`fi-decl-valo-koira`	4	58	232	58	232
Romanian	`ro-noun-f-ea`	23	10	230	10	230
Assamese	`as-proper noun`	58	7	406	7	406
Lower Sorbian	`dsb-decl-noun-17`	62	18	1116	18	1116
Gujarati	`*gu-conj-v`	1	7	7	-	0
Finnish	`*fi-decl-kala-koira`	2	63	126	-	0

Table 2: Randomly selected templates from random evaluation 3. Note that the last two templates have been removed after the evaluation process, and are noted with an asterisk.

Comparing our system's results to the most recent version of Kirov et al. *UniMorph* project corpus, it is noted that according to the available resources online, the *UniMorph* project currently has a corpus of 8.8M words, compared to our 6 to 6.5M words.[7] In addition, the *UniMorph* corpus has significantly more forms for high frequency languages, however, a lot of languages mentioned in the 'annotated languages' section do not have an available corpus (languages which are not available yet are listed in a different section in the webpage). As Table 3 demonstrates, *UniMorph* has many more forms for Arabic, but is lacking when it comes to low frequency languages such as the dialects of Alemannic German. Our system's lacking may occur either because the language experts for certain languages have created modules or badly-formatted templates to generate inflectional paradigms (and our system is not capable of processing either of those), but *UniMorph* has access to this information from accessing directly the lemmata's webpages.

Language		**UniMorph**		**Wikinflection**	
Name	***ISO***	***Forms***	***Paradigms***	***Forms before evaluation***	***Forms after evaluation***
Adyghe	ady	n/a	n/a	440	440
Albanian	sqi	33483	589	8767	8767
Alemannic German	gsw	0	0	232	232
Ancient Greek	grc	0	0	3312	3312
Arabic	ara	140003	4134	36	36
Aragonese	an	0	0	448	448
Armenian	hye	338461	7033	2824	59
Assamese	as	0	0	13790	13790
Asturian	ast	n/a	n/a	23599	23329
Avestan	ae	0	0	6	6
SUM		**8850395**	**309083**	**6518762**	**6024077**

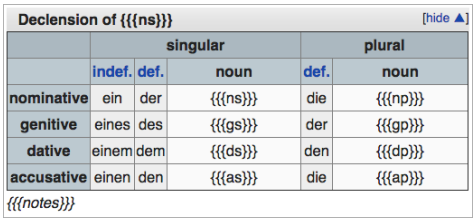
Table 3: A list of the fist 10 languages (in alphabetical order) and the number of their forms, from the *UniMorph* project and our system, *Wikinflection*. The full table is available online.[6] 'n/a' refers to languages that (according to the *UniMorph* project website) have been annotated, but no number has been published. '0' refers to the absence of the language from the corpus. For *Wikinflection* statistics, Random Evaluation No. 3 was used.

4 Discussion

Our approach to generate all inflectional paradigms from Wiktionary, on a large and multilingual scale, proved successful, but not as high-quality as we initially expected. First of all, we lost access to a big portion of inflectional information, because we only opted to use static template information and not any modules. Second, one of our greatest challenges was the different

[7] `https://unimorph.github.io/`, accessed November 21, 2018.

formatting of different languages; the dynamic links were different among languages as discussed in Section 3.3, and also the inflectional tables looked vastly different, with very inconsistent information flow, use of header rows, header columns and feature naming. This also caused loss of some morphological features when parsing inflectional tables, a problem which we are aware of and are in the process of solving. The inconsistencies span further in some cases, with templates which do not follow the Wiktionary-wide format of dynamic links (Figure 13a), or templates which are incomplete or contain links to other content, sometimes nonexistent (Figure 13b), and these templates were automatically rejected during parsing.

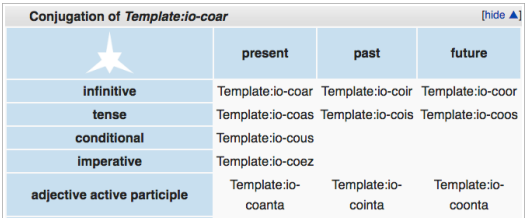


Declension of {{{ns}}} [hide ▲]

	singular			plural	
	indef.	def.	noun	def.	noun
nominative	ein	der	{{{ns}}}	die	{{{np}}}
genitive	eines	des	{{{gs}}}	der	{{{gp}}}
dative	einem	dem	{{{ds}}}	den	{{{dp}}}
accusative	einen	den	{{{as}}}	die	{{{ap}}}

{{{notes}}}

(a) Table from template `de-decl-noun-m`.



Conjugation of *Template:io-coar* [hide ▲]

	present	past	future
infinitive	Template:io-coar	Template:io-coir	Template:io-coor
tense	Template:io-coas	Template:io-cois	Template:io-coos
conditional	Template:io-cous		
imperative	Template:io-coez		
adjective active participle	Template:io-coanta	Template:io-cointa	Template:io-coonta

(b) Table from template `io-conj`.

Figure 13: Examples of templates which cannot be parsed by our system.

Another reason why we were only able to retrieve inflectional information from half of the lemmata with inflectional links was the ever-changing and evolving nature of the Wiktionary. Many entries contain inflectional links in the format of `{{rfinfl|LANGUAGE|POS}}`, which are actually placeholders for templates that do not exist yet[8]. Concerning the generated paradigms, because words of the same inflectional schema tend to follow similar morphological processes, we are confident to believe that the quality of the generated forms is at least satisfactory. There are cases of false negatives as discussed in Section 3.3, but these are to be expected from semi-supervised generation. False positives are rare, but may occur in cases where the inflectional paradigm requires more than one template or additional information from modules in order to be generated; for example, the template `{{hu-infl-nom}}`[9] used for nouns such as *'ring'* calls for extra parameters and server data in order to produce stem allomorphs during declension. While this approach is effective for generating the web tables, it impedes our generation of a correct inflectional paradigm, and also is inconsistent with the way the verb inflectional tables were made for the same language (e.g. `{{hu-conj-szem-üd}}`[10] requires a third parameter to produce stem allomorphs, an approach which could have been used for nouns too).

Despite the issues we had to overcome, our system is able to generate an inflectional corpus, with information which would be hard to extract even with state-of-the-art tools, such as stem allomorphs and affixes. This information is usually sparsely available, especially for low-resource languages such as Crimean Tatar, Võro and Northern Sami, and could prove to be a useful source not only for natural language processing, but also for linguistic research and language learners. In addition, with the improvement of our morphological feature tagging, we aim to create a large resource of tagged tokens and types, which could improve performance on many natural language processing tasks, especially those who require the use of low-frequency words.

[8] `https://en.wiktionary.org/wiki/Template:rfinfl`
[9] `https://en.wiktionary.org/wiki/Template:hu-infl-nom`
[10] `https://en.wiktionary.org/wiki/Template:hu-conj-szem-üd`

5 Conclusion

Wiktionary has become an essential linguistic resource, and it is important to ensure that all its available information is accessible for research. Our attempts to parse the Wiktionary for inflectional information have allowed us to utilize data which has either been partially available (Kirov et al. (2016), Liebeck and Conrad (2015)) or has not been available so far (stem allomorphs). Although we were only able to access a fraction of the available inflectional information, we were able to construct paradigms for over 140 languages, some of which being low-frequency languages and previously did not have available inflectional corpora. Our project is available online on Github[11] and can be downloaded and used alongside an English Wiktionary XML dump file, to produce a local corpus of inflectional paradigms. While we had to tackle several difficulties and we are currently in the process of perfecting the output, our system is a new approach to parsing and providing multilingual linguistic resources for computational morphology.

Our future work will focus, primarily, on improving template parsing so that all possible morphological features are extracted, and on performing human evaluation on the produced output in order to ensure high quality. We aim to keep increasing the size and quality of the generated corpora, by exploring whether the use of modules could be possible in some cases, and we would like to soon release the corpus as a pre-made resource as well, in order to be directly used. Additionally, we will explore how easy it would be to adapt our code to generate inflectional paradigms from other editions of Wiktionary – if the other editions maintain the same data and link structure as the English dump file, it could be as simple as translating a few headings and features in the code.

6 Acknowledgements

This work was partially funded by the European Union's Horizon 2020 grant agreement No. 731724 (iREAD).

[11] `https://github.com/lenakmeth/Wikinflection`

References

Acs, J., Pajkossy, K., and Kornai, A. (2013). Building basic vocabulary across 40 languages. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, pages 52–58, Sofia, Bulgaria. Association for Computational Linguistics.

Benko, V. (2016). Two years of aranea: Increasing counts and tuning the pipeline. In *LREC*.

Kirov, C., Sylak-Glassman, J., Que, R., and Yarowsky, D. (2016). Very-large scale parsing and normalization of wiktionary morphological paradigms. In *LREC*.

Liebeck, M. and Conrad, S. (2015). Iwnlp: Inverse wiktionary for natural language processing. 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)*, volume 2, pages 414–418.

MediaWiki (2018). Api:client code — mediawiki, the free wiki engine. [Online; accessed 1-October-2018].

Nivre, J., Abrams, M., Agić, Ž., Ahrenberg, L., Antonsen, L., Aplonova, K., Aranzabe, M. J., Arutie, G., Asahara, M., Ateyah, L., Attia, M., Atutxa, A., Augustinus, L., Badmaeva, E., Ballesteros, M., Banerjee, E., Bank, S., Barbu Mititelu, V., Basmov, V., Bauer, J., Bellato, S., Bengoetxea, K., Berzak, Y., Bhat, I. A., Bhat, R. A., Biagetti, E., Bick, E., Blokland, R., Bobicev, V., Börstell, C., Bosco, C., Bouma, G., Bowman, S., Boyd, A., Burchardt, A., Candito, M., Caron, B., Caron, G., Cebiroğlu Eryiğit, G., Cecchini, F. M., Celano, G. G. A., Čéplö, S., Cetin, S., Chalub, F., Choi, J., Cho, Y., Chun, J., Cinková, S., Collomb, A., Çöltekin, Ç., Connor, M., Courtin, M., Davidson, E., de Marneffe, M.-C., de Paiva, V., Diaz de Ilarraza, A., Dickerson, C., Dirix, P., Dobrovoljc, K., Dozat, T., Droganova, K., Dwivedi, P., Eli, M., Elkahky, A., Ephrem, B., Erjavec, T., Etienne, A., Farkas, R., Fernandez Alcalde, H., Foster, J., Freitas, C., Gajdošová, K., Galbraith, D., Garcia, M., Gärdenfors, M., Garza, S., Gerdes, K., Ginter, F., Goenaga, I., Gojenola, K., Gökırmak, M., Goldberg, Y., Gómez Guinovart, X., Gonzáles Saavedra, B., Grioni, M., Grūzītis, N., Guillaume, B., Guillot-Barbance, C., Habash, N., Hajič, J., Hajič jr., J., Hà Mỹ, L., Han, N.-R., Harris, K., Haug, D., Hladká, B., Hlaváčová, J., Hociung, F., Hohle, P., Hwang, J., Ion, R., Irimia, E., Ishola, Ọ., Jelínek, T., Johannsen, A., Jørgensen, F., Kaşıkara, H., Kahane, S., Kanayama, H., Kanerva, J., Katz, B., Kayadelen, T., Kenney, J., Kettnerová, V., Kirchner, J., Kopacewicz, K., Kotsyba, N., Krek, S., Kwak, S., Laippala, V., Lambertino, L., Lam, L., Lando, T., Larasati, S. D., Lavrentiev, A., Lee, J., Lê Hồng, P., Lenci, A., Lertpradit, S., Leung, H., Li, C. Y., Li, J., Li, K., Lim, K., Ljubešić, N., Loginova, O., Lyashevskaya, O., Lynn, T., Macketanz, V., Makazhanov, A., Mandl, M., Manning, C., Manurung, R., Mărănduc, C., Mareček, D., Marheinecke, K., Martínez Alonso, H., Martins, A., Mašek, J., Matsumoto, Y., McDonald, R., Mendonça, G., Miekka, N., Misirpashayeva, M., Missilä, A., Mititelu, C., Miyao, Y., Montemagni, S., More, A., Moreno Romero, L., Mori, K. S., Mori, S., Mortensen, B., Moskalevskyi, B., Muischnek, K., Murawaki, Y., Müürisep, K., Nainwani, P., Navarro Horñiacek, J. I., Nedoluzhko, A., Nešpore-Bērzkalne, G., Nguyễn Thị, L., Nguyễn Thị Minh, H., Nikolaev, V., Nitisaroj, R., Nurmi, H., Ojala, S., Olúòkun, A., Omura, M., Osenova, P., Östling, R., Øvrelid, L., Partanen, N., Pascual, E., Passarotti, M., Patejuk, A., Paulino-Passos, G., Peng, S., Perez, C.-A., Perrier, G., Petrov, S., Piitulainen, J., Pitler, E., Plank, B., Poibeau, T., Popel, M., Pretkalniņa, L., Prévost, S., Prokopidis, P., Przepiórkowski, A., Puolakainen, T., Pyysalo, S., Rääbis, A., Rademaker, A., Ramasamy, L., Rama, T., Ramisch, C., Ravishankar, V., Real, L., Reddy, S., Rehm, G., Rießler, M., Rinaldi, L., Rituma, L., Rocha, L., Romanenko, M., Rosa, R., Rovati, D., Roșca, V., Rudina, O.,

Rueter, J., Sadde, S., Sagot, B., Saleh, S., Samardžić, T., Samson, S., Sanguinetti, M., Saulīte, B., Sawanakunanon, Y., Schneider, N., Schuster, S., Seddah, D., Seeker, W., Seraji, M., Shen, M., Shimada, A., Shohibussirri, M., Sichinava, D., Silveira, N., Simi, M., Simionescu, R., Simkó, K., Šimková, M., Simov, K., Smith, A., Soares-Bastos, I., Spadine, C., Stella, A., Straka, M., Strnadová, J., Suhr, A., Sulubacak, U., Szántó, Z., Taji, D., Takahashi, Y., Tanaka, T., Tellier, I., Trosterud, T., Trukhina, A., Tsarfaty, R., Tyers, F., Uematsu, S., Urešová, Z., Uria, L., Uszkoreit, H., Vajjala, S., van Niekerk, D., van Noord, G., Varga, V., Villemonte de la Clergerie, E., Vincze, V., Wallin, L., Wang, J. X., Washington, J. N., Williams, S., Wirén, M., Woldemariam, T., Wong, T.-s., Yan, C., Yavrumyan, M. M., Yu, Z., Žabokrtský, Z., Zeldes, A., Zeman, D., Zhang, M., and Zhu, H. (2018). Universal dependencies 2.3. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Ricco, J. (2017). Using python to scrape html tables with merged cells. [Online; accessed 7-October-2018].

Roland, O. (2011). Dictionary builder. https://github.com/newca12/dictionary-builder. [Online; accessed 1-October-2018].

Wikipedia contributors (2017). Wiktionary:parsing. [Online; accessed 1-October-2018].

Zesch, T., Müller, C., and Gurevych, I. (2008). Extracting lexical semantic knowledge from wikipedia and wiktionary. In *LREC*, volume 8, pages 1646–1652.

Preprocessing Does Matter: Parsing Non-Segmented Arabic

Noor Abo Mokh, Sandra Kübler
Indiana University, Bloomington, IN
noorabom@iu.edu, skuebler@indiana.edu

ABSTRACT
Preprocessing is a normal first step in parsing, but it is the step that most researchers consider trivial and not worth reporting. The problem is exacerbated by the fact that parsing research often focuses on parsing a treebank rather than parsing a text since the treebank obscures many of the preprocessing steps that have gone into the curation of the text. In this paper, we argue that preprocessing has a non-negligible effect on parsing, and that we need to be careful in documenting our preprocessing steps in order to ensure replicability. We focus on parsing Arabic since Arabic is more difficult than English in the sense that 1) the orthography has intricacies such as vocalization that need to be handled and that 2) the basic units in the treebank do not necessarily correspond to words but sometimes constitute morphemes. The latter necessitates the use of a segmenter in order to convert the text to a form that the parser has seen in training. We investigate a scenario where we combine a morphological analyzer/segmenter, MADAMIRA, with a parser trained on the Arabic Treebank. We mainly examine the differences in orthographic and segmentation decisions between the analyzer and the treebank. We show that normalizing the two representations is not a simple process and that results can be artificially low or misleading if we do not pay attention. In other words, this paper is an attempt at establishing best practices for parsing Arabic, but also more generally for documenting preprocessing more carefully.

KEYWORDS: Parsing, Arabic, preprocessing.

1 Introduction

Preprocessing is a normal step before parsing, it encompasses all the steps to convert the text to be parsed so as to model the decisions in the treebank that was used to train the parser. Details of preprocessing are generally not reported in the parsing literature because this step is considered trivial and not part of the research. Also, in parsing Arabic literature, gold segmentation of the text to be parsed is usually assumed. This means that that we ignore the issues of converting the text to be parsed to the format seen in traning the parser. However, the decisions taken in preprocessing are of vital importance to ensure parsability and can have major effects on parsing if we are not careful.

Our focus is on parsing Arabic, and our ultimate goal is research on integrating segmentation decisions jointly with the parsing step by presenting the different possible segmentations for words in a lattice to the parser. Such a setting is more representative of naturally occurring data, as no gold segmentation is assumed.

However, in working with the state of the art morphological analyzer/segmenter for Arabic, MADAMIRA (Pasha et al., 2014), we found that there are differences in segmentation decisions and orthographic normalizations between the treebank and the segmenter (for details see section 2) . These need to be addressed before we can successfully combine the segmenter and the parser. This paper describes the non-trivial process of finding a common representation. If these issues are not addressed carefully, parsing results will be artificially low or severely misleading.

All of the issues addressed in this work are considered preprocessing, and to our knowledge are not described in publications on Arabic parsing in enough detail to allow for replicability. Thus this paper is intended as an attempt at establishing best practices for parsing Arabic, but also more generally for documenting preprocessing carefully enough to ensure replicability. Note that these issues do not occur when we focus on parsing treebank data since we then implicitly use the same (potentially undocumented) preprocessing used in the treebank.

The remainder of this paper is organized as follows: Section 2 presents a closer inspection of the issues that we address in the following sections. In section 3, we show how preprocessing has been handled and documented in previous work, or how published work is often short on details. In section 4, the experimental design is described, including a description of the treebank and tools used in this paper. Section 5 explains the process of adjusting MADAMIRA analyses to mirror the treebank decisions and shows the effects that the individual steps have on parsability. We conclude by a discussion of the bigger picture of preprocessing for parsing Arabic in section 6.

2 Preprocessing Issues

When we envision the use of a parser in a realistic scenario, the parser needs to handle standard text, split into sentences. However, parsing research generally focuses on parsing data from treebanks, which has already been preprocessed. For English, the discrepancy between 'realistic' text and treebank text is not dramatic, it mostly consists of splitting contractions such as "you'll" or "can't".

In languages such as Arabic, the difference is much greater, for a range of reasons. One of the differences concerns the fact that in Arabic, clitics such as pronouns and prepositions are agglutinated as affixes to the subsequent word. In the treebanks, these clitics tend to be split

off and treated as separate words, thus necessitating a segmentation step before we can parse new text. However, the problem is complicated by the fact that the segmentation of a surface word may be context dependent and can thus only be reliably performed during parsing. For example, the word الَام "AlAm" [1] can be segmented in three different ways, with very different meanings and morpho-syntactic structure: الـ أم "Al >m" (Eng.: the mother), آلَام "|lAm" (Eng.: pains), أ لَام "> lAm" (Eng.: did he blame?). Another issue, which is intricately connected to the segmentation step, concerns orthographic normalization, for example, vocalization and Hamza normalization. Arabic is normally written without short vowels and some Hamza variants, but the treebank may contain those because they often disambiguate the word. Also, the different spelling variants, e.g., of the Hamza, are sometimes inconsistently used by Arabic users (where some writers may ignore the Hamzas or use a different Hamza variant). Segmenters and treebanks may decide to normalize to one Hamza variant, but not necessarily to the same one.

The sentence in (1) shows an example of the variation between original text, MADAMIRA (Pasha et al., 2014) analyses, and the representation in the Penn Arabic Treebank[2] (Seddah et al., 2013), with all the phenomena discussed above. The source text is not segmented, MADAMIRA provides automatic segmentation, and the Arabic Treebank provides gold segmentation.

(1) a. Arabic script:

الجغرَافي لتُؤكد نظريتهَا

Eng.: 'the geographic to emphasize her theory'

b. Source text: **AljgrAfy** lt&kd **nZrythA**
c. Treebank: Al**jgrAfy** l t&kd **nZryt hA**
d. MADAMIRA: Al **jgrAfy** l t&kd **nZryp hA**

When we compare the three versions, we expect the difference in segmentation between the source text on the one hand and MADAMIRA and treebank on the other hand. However, we also see additional differences: First, the treebank and MADAMIRA make different segmentation decisions: While MADAMIRA splits of the definite article "Al" in الجغرَافي "AljgrAphy" (Eng.: geographic), the treebank does not. Second, the feminine marker representation in the treebank (نظريتـ, "nZryt", Eng.: theory) does not match with the representation in MADAMIRA (نظرية, "nZryp").

To the best of our knowledge, these decisions are not documented fully, but they need to be handled in preprocessing if we want to successfully parse text that was segmented by MADAMIRA.

3 Related Work

3.1 Preprocessing

Preprocessing can concern issues of orthographic normalization, segmentation, the data splits, and converting trees into training data. While we are only concerned with the first two issues, all such steps need to be reported: Dakota and Kübler (2017) have shown that decisions in

[1] We use Buckwalter transliteration (Buckwalter, 2004) throughout the paper since this is also used in the treebank and segmentation step.

[2] In the version used in the SPMRL shared tasks 2013/2014.

preprocessing can have a significant impact on parsing results. A study by Goodman (1996) shows that he was not able to duplicate the results of Bod (1996) because Bod did not give sufficient details on either the preprocessing of the data or the data split. Also, Cheung and Penn (2009) report that they were not able to replicate experiments by Becker and Frank (2002) because Becker and Frank performed manual correction which were not documented.

Given this situation, a detailed description of all preprocessing steps is indispensable to enable replicability of such studies.

There are only very few parsing papers that broached the topic explicitly: Wagner et al. (2007) detail the preprocessing steps performed on the data from the BNC to match the Penn Treebank representation. Seddah et al. (2012) describe preprocessing steps such as normalization, style modifications, spell corrections, for adaptation of noisy user-generated content to a Penn treebank-based parser.

However, in much of the current parsing literature, preprocessing is minimally described, if it is mentioned at all. The lack of preprocessing information can be found in parsing studies for many languages. For English, Bod (2001) describes that all trees were stripped off their semantic tags, coreference information, and quotation marks but no other details are mentioned. Charniak and Johnson (2005) mention that they used "the division into preliminary training and preliminary development data sets described in (Collins and Koo, 2005)" but no other preprocessing details are provided. In (Collins and Koo, 2005), only information on the data split is mentioned and no other preprocessing steps are documented, this is also the case for work by McClosky et al. (2006). Klein and Manning (2003) mentions the tree annotation and transformation were similar to the ones described by Johnson (1998). While Johnson (1998) details his method of tree transformation, only the data splits are mentioned. Petrov and Klein (2007), work on English, Chinese and German, they do not provide information about preprocessing.

3.2 Preprocessing Arabic

Preprocessing steps in parsing Arabic studies is also minimally described which makes it difficult to duplicate results of such studies. Most of the studies on Arabic parsing, specifically in parsing of Modern Standard Arabic (MSA) studies, use treebank data where gold segmentation is usually assumed. Preprocessing of such text is minimally described, as it is often presumed that the data sets are similar, i.e., they use matching segmentation schemes and preprocessing steps in the training data and in the test data.

In their work on dialectal Arabic, Chiang et al. (2006) describe the data split in addition to the tree transformations, they also mention that they use the undiacriticized form. Al-Emran et al. (2015),i n a study on parsing MSA, also used treebank data, but no other details on preprocessing are mentioned. In (Attia et al., 2010), only data splits are mentioned, but no details about the preprocessing steps. In the SPMRL shared task description (Seddah et al., 2013, 2014), the specifics of normalization are described: "all tokens are minimally normalized: no diacritics, no normalization except for the minimal normalization present in MADA's back-end tools". It is also mentioned that they follow the standard ATB segmentation, where categories of orthographic clitics are split except from the article Al+.

Other studies use an external tool for segmentation and tokenization. Kulick and Bies (2009) used a morphological analyzer, but no preprocessing steps were documented, only the data split is mentioned. However, they mention that since there is an issue of mismatching tokens, they

did not report a parsing score, because of the evaluation method. I.e., the tokens do not match because of differences in segmentation schemes used, hence, the constituent spans cannot be compared.

Green and Manning (2010) experiment with two scenario, one where gold segmentation is assumed, and one where a joint segmentation and parsing experiment is performed. In the gold scenario, preprocessing steps mentioned are removing all diacritics, normalizing Alif variants, mapping the Arabic punctuation marks to their Latin equivalents, and segmentation markers were kept. They used this normalization because other orthographic normalization schemes suggested by Habash and Sadat (2006) which they experimented with, had an insignificant influence on parsing performance in comparison to their normalization scheme. In their non-gold scenario, they experiment with two different pipelines, one where they use a manually-created list of clitics provided by Maamouri et al. (2004), to generate the lattices. They mention that no other preprocessing was made for this setting besides a correction rule for a deleted 'A' from a determiner 'Al'. This pipeline was compared to another pipeline in which Green and Manning (2010) used MADA (v3.0) to generate 1-best analyses. No other information or details are mentioned about preprocessing.

Another question which often remains unclear concerns the issue that in naturally occurring data, some diacritics or Hamza variants might be present in the text to be parsed. Therefore, it is unclear how much normalization is needed. This was also addressed by Maamouri et al. (2008), who show that while non-diacritized forms are the default, it might not be representative of real world data.

4 Experimental Setup

4.1 Treebank

For our experiments, we use the Penn Arabic Treebank (PATB) (Maamouri et al., 2004) from the 2013 SPMRL Shared Task (Seddah et al., 2013, 2014). The shared tasks provided constituent trees and dependency trees, the latter based on the annotations of the Columbia Arabic Treebank (CATiB) (Habash and Roth, 2009). The two versions are aligned at the word level.

For our experiments, we use the constituent version. The training data we use are unvocalized, except from few cases of Hamzas (considering that Hamza is a form of diacritization).

For training, we use the 5k dataset (i.e., the first 5 000 sentences of the complete training file), and we use the dedicated test set containing 1 959 sentences.

4.2 MADAMIRA

To segment the original text, we use the morphological analyzer MADAMIRA (Pasha et al., 2014). MADAMIRA is a morphological analyzer for Arabic (Pasha et al., 2014), a combination of MADA (Habash et al., 2009; Habash and Rambow, 2005) and AMIRA (Diab et al., 2004; Diab, 2009), which performs segmentation as part of the morphological analysis.

Text in Arabic script is transliterated into Buckwalter transliteration (Buckwalter, 2004). The morphological analyzer produces a set of possible analyses for each word. This list is then processed by the feature modeling component, where an SVM and language models are applied to create predictions of morphological features and other features such as diacritics for each token. The list is ranked based on the model predictions, and the text is tokenized based on morphological features. For our current work, we use only the highest ranked analyses.

4.3 Parser

We use the *Blatt Parser* (Goldberg and Elhadad, 2011), a reimplementaiton of the Berkeley Parser (Petrov et al., 2006; Petrov and Klein, 2007) which was extended to allow lattices as input. We use five split-merge cycles for training.

4.4 Evaluation

For the evaluation, we use the scorer from the SPMRL 2013 Shared Task (Seddah et al., 2013). The scorer is derived from the *EVALB* version used for the SANCL 2012 "Parsing the Web" shared task (Petrov and McDonald, 2012), which in turn is based on teh version by Sekine and Collins (1997); with additional options, such as allowing the evaluation of function labels and penalizing unparsed sentences. EVALB, in contrast, reports the number of unparsed sentences. However, they are ignored in the calculation of precision and recall.

For the purposes of the work reported here, we are not interested in the standard evaluation metrics. Instead, we focus on the errors report, i.e., the cases flagged by the evaluation script as containing errors. Such errors can consist of differences in the words in the parsed text and the treebank, rather than in the tree structures. The number of mismatches serves as an indicator of how different the MADAMIRA analyses are from the treebank sentences. We do not use the standard evaluation metrics because we are interested in the differences in spelling and segmentation between the parser output and treebank sentences, which can be found in the errors report. Looking at the parsing results may not be representative of the actual results if many sentences are ignored because of such differences. For instance, assuming only 100 sentences were parsed without errors out of the 1 959 sentences, and the score is 97.2, then this is inaccurate as the score does not reflect the errors found in sentences.

In the errors report, there are two types of error messages; one flags sentences where there is a mismatch in sentence length, i.e., the number of tokens in the sentence segmented by MADAMIRA does not match the number of tokens in the treebank. Mismatch in length can be as a result of segmentation differences. For instance, the phrase الجولة السَّادسة "Aljwlp AlsAdsp" (Eng.: the sixth round), from the treebank is analyzed as "Al jwlp Al sAdsp". Splitting off the determiner 'Al' is not an error, but is a case where MADAMIRA and the treebank made different decisions.

The second type of errors flags tokens which do not match with their corresponding tokens in the treebank. We see this type of errors occurring because of differences in orthographic normalization, for example in Hamza representation: MADAMIRA returns إنتظر "{ntZr" (Eng.: waited), which is represented as انتظر "AntZr" in the treebank; i.e., the treebank normalizes the Hamza { to A.

Note that there is an interdependence between the two error types since token mismatches are only reported if the number of tokens matches between parser output and treebank. This means that when we correct a segmentation difference, such as the split-off determiner described above, the number of length mismatches will decrease, but the number of token mismatches may increase since the corrected sentences are now checked for token mismatches.

5 Preprocessing Steps

As described above, our ultimate goal is to create different segmentation hypotheses using MADAMIRA and then make final segmentation decisions during parsing. Our current work

	Normalization	# length errors	# token errors/diff.	# no error	# skipped sent.
1	Baseline	1 899	50	3	7
2	+ delete diacritics	1 900	16	37	6
3	+ fixed determiner	582	836	541	0
4	+ modified fem. pronouns	582	795	582	0
5	+ modified numbers	469	865	625	0

Table 1: First steps of normalization and their effect on parsability.

focuses on the preprocessing steps necessary to create an interface between MADAMIRA and the Arabic Treebank representations. This means that we need to determine the differences in representations and normalize them automatically as far as possible. We assume that there will be some differences that are too idiosyncratic in nature to be handled automatically and would need manual intervention. We use the SPMRL scorer to find all differences. However, the procedure is complicated by the fact that some of the differences will result from incorrect segmentation. These latter differences concern our ultimate goal and will not be addressed here since we need to make sure that we handle all the preprocessing decisions before we address the problem of segmentation.

In the following sections, we describe the baseline and all the modifications along with their effect on the evaluation. An overview of these effects can be found in table 1.

5.1 Baseline Experiment

The baseline experiment simulates the case where we present MADAMIRA with standard text (in Buckwalter transliteration). We use MADAMIRA's analyses as input for the parser, with only one modification: The standard text from the treebank contains the symbols -LBR- and -RBR- instead of opening and closing parentheses, which we changed back to the original parentheses. The reason why the parentheses in the text were replaced is that many parsers cannot distinguish between parentheses as part of the sentence and syntactic brackets, which are also marked by parentheses. MADAMIRA, in contrast, expects parentheses, not the replacement symbols.

MADAMIRA provides for each word its diacriticized form, part of speech, Buckwalter transliteration, the lemma, and morphological information. For the purpose of this experiment, segments of a diacriticizised token were extracted. For instance, لمُؤتمر "lm&tmr" (Eng.: to a conference), is segmented into 'li' ل and 'mu&otamari', where 'li' is a preposition, and 'mu&otamari' is a noun. Diacritics were kept in the representation of each token for the baseline experiment since ultimately we need to extract all possible segmentations, which are only available in the diacriticized form from MADAMIRA. Based on this experiment, only three sentences did not cause any errors in the evaluation script. I.e., all other sentences could not be evaluated because of mismatches on the word level. This was not unexpected, as diacritics are not included in the treebank[3]. However, this result makes it very obvious that we need to adapt MADAMIRA analyses before parsing.

5.2 Deleting Diacritics

The obvious next step is to delete diacritics. No other modifications are performed on the data. Based on this setting, the scorer reports 1 916 errors, 1 900 of which concern sentences with

[3]Here we refer to the dataset used in this study, other releases of the treebank data include diacritized forms

length mismatches (see line 2 in table 1). The number of sentences without errors increases from 3 to 40, which is a much lower improvement than expected, indicating that there are additional, severe problems in the sentences.

5.3 Handling the Definite Article 'Al'

Since removing the diacritics did not improve results significantly, the next step is to determine other causes for length and thus represenetational mismatches. Looking through the sentences that were flagged for length mismatch, we found an issue concerning the segmentation of the determiner 'Al' in noun phrases. In written forms in Arabic, the determiner is usually connected to the noun or adjective, forming one orthographic unit. But in MADAMIRA's analyses, the determiner and noun or adjective are segmented into two different tokens, as described in section 4.4. This is different from the treebank, where the determiner and noun or adjective occur as one orthographic unit. For instance, the Arabic word التّفّاحة "AltfAHp" (Eng.: apple) is represented as two segments in MADAMIRA, 'Al' ال, the determiner, and 'tfAHp', the noun. In the treebank, this word is represented as "Al+tfAHp", with determiner and noun forming a single token. This difference results in a length mismatch in the evaluation script.

After reattaching the determiner, the number of sentences with length mismatches decreases dramatically from 1 900 to 582, and the number of sentences without errors increases from 40 to 541 (see line 3 in table 1). However, the number of token mismatches increased considerably as well, to 836. This means that many of the sentences that do not have a length mismatch anymore still have a token mismatch and thus cannot be evaluated.

5.4 Feminine Marker Ta-Marbuta

Another look at the data shows that out of the 836 token mismatch cases, there are 138 sentences involving the feminine marker Ta-Marbuta. Overall, there are over 300 cases with a mismatch of the Ta-Marbuta in those sentences. The feminine marker Ta-Marbuta 'p' is transliterated as 't' in the treebank if it is followed by a pronominal clitic. In MADAMIRA's analyses, however, the modification of the Ta-Marbuta remains 'p', even if followed by a clitic. For instance, كتابته "ktAbth" (Eng.: his writing) is segmented by MADAMIRA as كتابة ه "ktAbp h" while it is represented as كتابتـ ه "ktAbt h" in the treebank.

Therefore, we change the transliteration of the feminine marker when it is followed by a pronoun in MADAMIRA's analyses to match the representation in the treebank. The evaluation results show that the number of length mismatches does not change. This is to be expected since the Ta-Marbuta normalization does not change the segmentation, but only the word-internal representation. The number of token mismatches decreases slightly, from 836 to 795 (see line 4 in table 1). In addition, the number of sentences without errors increases from 541 to 582. It is worth noting that the decrease in token mismatches does not match the number of sentences displaying this issue. This means that some of the sentences in which the Ta-Marbuta was normalized still contain other mismatches.

5.5 Numbers representation

In the next step, when checking the texts again, we found another frequent difference in the representation of numbers separated by a hyphen. As an example, in the treebank, hyphenated

Hamza Type	# in MADAMIRA	# in Treebank
lone Hamza (')	1 017	1 017
Hamza on Wa (&)	417	417
Hamza on Ya (})	1 303	1 303
Hamza above Alif (>)	6 140	2 156
Hamza below Alif (<)	2 683	917
Alif Al-Wasla ({)	1 577	0
Madda above Alif (|)	217	173
Alif Maqsura (Y)	2 379	2 447

Table 2: Hamza types in MADAMIRA and the Penn Arabic Treebank.

numbers are represented as '2-1'; however, in MADAMIRA, these were segmented into '2 - 1'.

After modifying the number representations, the number of length mismatches decreases (see line 5 in table 1). This is expected, given the fact that the former segmentation of numbers affect the length of sentences, hence resulting in such errors.

5.6 Hamza and Alif Inconsistencies

Besides the mismatches described above, there are issues involving the Hamza and Alif representation. Alif is a letter while Hamza is a diacritic-like letter mark (Habash, 2010). I.e., the Hamza has similarities with a diacritic, but Habash notes that "The general consensus on encoding the Hamza is to consider it a letter mark (and as such part of the letter) as opposed to being a diacritic". The Hamza can appear on letters such as Alif, Wa and Ya. The form of the Hamza varies depending on its position in the word.

Hamza inconsistencies can be due to inconsistencies in spelling in Arabic texts rather than normalization efforts during segmentation and analysis since Arabic writers often do not write the Hamza or use different variants of Hamza, thus making it an optional mark. This is the reason why variants of Hamza that occur with the letter Alif are usually normalized.

Examining the segmentation in MADAMIRA and the treebank text, there seem to be differences in the spelling choices. Table 2 shows the frequencies of Hamza and Alif variants in both variants. While some Hamzas (lone, on Wa, and on Ya) match in numbers in the treebank and in MADAMIRA, it seems that <, > and { (an alif variant) cause most of the inconsistencies, as indicated by the differences in counts between the two representations.

Due to the inconsistencies in Hamza and Alif spelling, we investigate two approaches: We first modify the Hamza representation in the MADAMIRA analyses such that specific variants of Hamzas are changed to match the treebank representations. In case the modification cannot be performed automatically, we use the second approach, which is a more traditional approach, where Hamza variants are normalized to 'A' in MADAMIRA in one experiment, and in both MADAMIRA and treebank in the other experiment.

In the first approach, we extract all tokens where Hamza occurs and compare them to the gold dataset representation. Based on the counts of Hamzas in table 2, the most serious discrepancy concerns the Alif Al Wasla '{' because it is only used in MADAMIRA but not in the treebank. Since it is not clear what the best normalization for the Alif Al Wasla to approximate the treebank representation of the concerned words, we perform three different experiments where only

Normalization	# length errors	# token errros	# no error
result from table 1	469	865	625
Alif Al-Wasla → Hamza below Alif	469	865	625
Alif Al-Wasla → Hamza above Alif	469	862	628
Alif Al-Wasla → A	469	391	1099
Hamza above Alif → A	469	1 314	176
Hamza above Alif → Hamza below Alif	469	1 309	181
Normalized Hamza in MADAMIRA	469	1 352	138
+ in treebank	469	229	1261

Table 3: Hamza Modification following two different approaches.

specific variants of Hamza are are used, to see which nromalization would give us the closest match to the treebank representation.

The Alif Al-Wasla '{' representation was changed to three different possible normalized forms: 1) Hamza below Alif <, 2) Hamza above Alif >, and 3) bare Alif 'A'. Additionally, we investigated whether the Hamza above Alif should be normalized to a bare Alif 'A' or to Hamza below Alif '<'.

The second approach, which is often the traditional way of addressing Hamza spelling inconsistencies, is normalization of Hamza variants to bare 'A'. The Hamzas that are normalized are: lone Hamza, Madda on Alif, Hamza above and below Alif, Alif Al-Wasla. Hamza on Wa and Ya are not normalized, because their counts are similar in both datasets (as shown in Table 3). Also, normally such Hamzas are kept in writing. In one experiment, Hamza variants were normalized in the training set and in MADAMIRA analyses. The second experiment included also normalization of Hamza variants in the treebank.

The results are shown in table 3. The best results of the first method were achieved when changing Alif Al-Wasla '{' to bare Alif 'A'. This is for two reasons: First, '{' is not represented in the treebank. Second, in many cases where '{' is used in MADAMIRA, 'A' is used in treebank.

Trying to normalize the Hamza above Alif either to A or to the Hamza below Alif increases the number of token errors, i.e., is not a feasible normalization. For the second set of experiments, where all Hamza instances were normalized (excluding Hamza on Wa or Ya), we found that we need to normalize the Hamza in both MADAMIRA analyses and in the treebank in order to obtain a good normalization. This normalization results in the highest number of sentences without errors, 1 261 out of 1 959. However, note that this is a somewhat radical step since it involves modifying the gold standard.

5.7 Remaining Issues

The results in Table 3 show that we can reduce the number of both types of errors/differences considerably. However, we did not manage to reduce the number of length or tokens mismatches in all cases. This was expected: The remaining errors are a result of segmentation differences. Most of these segmentation differences cannot be solved by normalization or simple modifications. Instead, they require syntactic analysis in combination with morphological analysis. For instance, the prepositional phrase لَّاجىء "llAjY" (Eng. to a refugee, or to the refugee) was represented as لَاجىء ال ل "l Al lAjY" (Eng.: to the refugee) in MADAMIRA, while based on the gold segmentation it is "l lAj}" (ENG: to a refugee) (note the difference between the definite

and indefinite forms). In this case, more information is needed to determine the correct form (whether it is an iDafa construction or a noun-adjective construction).

Another issue we also encountered concerns the Alif Maqsura (Y), a variant of Alif, whose spelling changes if followed by an affix to either 'y' or 'A'. The problem we encountered in this case was that a word like علَى "ElY" (Eng.: on) is represented as علِيّ "Ely" in MADAMIRA, but as "ElY" in the treebank if followed by a suffix. One way to overcome this issue is normalizing all instances of Alif Maqsura. However, this will create more errors since there were other instances where Alif Maqsura 'Y' is used for adjectives in the treebank, but MADAMIRA would analyze them as nisba adjective ending (i.e., -y). Normalizing all instances of 'Y' would result in errors for instances were the correct analysis is predicted by MADAMIRA. For this reason, we did not normalize the Alif Maqsura.

There are additional differences that need manual modifications, especially in the case of proper nouns (for instance, when the proper nouns starts with 'f' or 'k' which resemble clitics), or inconsistencies in segmentation between the treebank text and MADAMIRA analyses. For instance, the word لَالَّا "l{lA" (Eng.: so that not), in the MADAMIRA analysis is represented as لـ أن لَا "l >n lA", i.e., the word is segmented, causing a token mismatch. The same happens for the word لكي "lky" (Eng.: so) in MADAMIRA, which is represented as "l ky" in the treebank. These are idiosyncratic differences that need to be handled on a word by word basis. There are also few misspelled words in the source text that are not recognized by MADAMIRA.

While we used normalization, and found it to be the best option to reduce the number of sentences without errors, we believe the case of Hamza should be handled differently. This was also examined by Maamouri et al. (2008), who show that normalizing the dataset is not enough.

6 Conclusion

The work described in this paper describes a situation where we parse Arabic in a realistic setting, i.e, where we do not assume treebank segmentation, but instead use a state-of-the–art segmenter and morphological analyzer to obtain segmentation. Since we need a graph of all possible analyses for our future research, we need to use the vocalized forms without being able to use specific tokenization schemes that MADAMIRA offers. The numbers presented in section 4.4 show very clearly that we need to be extremely careful in how we handle differences in orthography and segmentation. These differences are partly due to segmentation ambiguities that cannot be resolved without access to syntactic information. These are the issues that we were originally interested in. However, such cases cases are overshadowed by a wide range of other cases, which need to be resolved in order to be able to focus on the interesting cases. The cases that need to be handled before we can even seriously start thinking about parsing experiments including different decisions in segmentation such as the case of the definite determiner and the Ta-Marbuta, but also orthographic inconsistencies involving Hamza and Alif. All of those decisions need to be explained in detail in order to ensure that the parsing research is replicable.

For the future, we plan to investigate a lattice approach to parsing. In this setting, the parser will be provided with a set of analyses from MADAMIRA, integrated into a lattice, which will allow the parser to choose the segmentation that works best given the syntactic constraints

of the grammar. This opens the question of how many analyses we should give the parser, since having access to too many irrelevant analyses may be more of a hindrance than helpful. This was shown in work on integrating word segmentation and parsing for Chinese (Hu et al., 2017). Other issues concerns the weighting of the arcs in the lattice, which have been shown to be useful in Arabic lattice parsing (Green and Manning, 2010), the usefulness of automatic morphological analyses, and the effect of normalization on parsing results.

Acknowledgments

We are grateful to Djamé Seddah for giving us his feedback on the experiments and the interpretation of the results. We are also grateful to the anonymous reviewers for their insightful comments.

References

Al-Emran, M., Zaza, S., and Shaalan, K. (2015). Parsing Modern Standard Arabic using treebank resources. In *2015 International Conference on Information and Communication Technology Research (ICTRC)*, pages 80–83.

Attia, M., Foster, J., Hogan, D., Roux, J. L., Tounsi, L., and Van Genabith, J. (2010). Handling unknown words in statistical latent-variable parsing models for Arabic, English and French. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 67–75. Association for Computational Linguistics.

Becker, M. and Frank, A. (2002). A stochastic topological parser for German. In *Proceedings of the 19th Onternational Conference on Computational Linguistics*, pages 1–7.

Bod, R. (1996). Monte Carlo Parsing. In Bunt, H. and Tomita, M., editors, *Recent Advances in Parsing Technology*, pages 255–280. Kluwer.

Bod, R. (2001). What is the minimal set of fragments that achieves maximal parse accuracy? In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 66–73.

Buckwalter, T. (2004). Arabic morphological analyzer version 2.0. Linguistic Data Consortium.

Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 173–180.

Cheung, J. C. K. and Penn, G. (2009). Topological field parsing of German. 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 64–72.

Chiang, D., Diab, M., Habash, N., Rambow, O., and Shareef, S. (2006). Parsing Arabic dialects. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

Collins, M. and Koo, T. (2005). Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

Dakota, D. and Kübler, S. (2017). Towards replicability in parsing. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 185–194, Varna, Bulgaria.

Diab, M. (2009). Second generation AMIRA tools for Arabic processing: Fast and robust tokenization, POS tagging, and base phrase chunking. In *2nd International Conference on Arabic Language Resources and Tools*, volume 110.

Diab, M., Hacioglu, K., and Jurafsky, D. (2004). Automatic tagging of Arabic text: From raw text to base phrase chunks. In *Proceedings of HLT-NAACL 2004: Short papers*, pages 149–152.

Goldberg, Y. and Elhadad, M. (2011). Joint Hebrew segmentation and parsing using a PCFG-LA lattice parser. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 704–709.

Goodman, J. (1996). Efficient algorithms for parsing the DOP model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA.

Green, S. and Manning, C. D. (2010). Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 394–402.

Habash, N. (2010). *Introduction to Arabic Natural Language Processing*, volume 3 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers.

Habash, N. and Rambow, O. (2005). Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 573–580.

Habash, N., Rambow, O., and Roth, R. (2009). MADA+ TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In *Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, volume 41, Cairo, Egypt.

Habash, N. and Roth, R. M. (2009). Catib: The Columbia Arabic Treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference*, pages 221–224.

Habash, N. and Sadat, F. (2006). Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 49–52.

Hu, H., Dakota, D., and Kübler, S. (2017). Non-deterministic segmentation for Chinese lattice parsing. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, Varna, Bulgaria.

Johnson, M. (1998). PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430.

Kulick, S. and Bies, A. (2009). Treebank analysis and search using an extracted tree grammar. In *Eighth International Workshop on Treebanks and Linguistic Theories*.

Maamouri, M., Bies, A., Buckwalter, T., and Mekki, W. (2004). The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, volume 27, pages 466–467, Cairo, Egypt.

Maamouri, M., Kulick, S., and Bies, A. (2008). Diacritic annotation in the Arabic treebank and its impact on parser evaluation. In *LREC*.

McClosky, D., Charniak, E., and Johnson, M. (2006). Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 337–344.

Pasha, A., Al-Badrashiny, M., Diab, M. T., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. (2014). MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *LREC*, volume 14, pages 1094–1101.

Petrov, S., Barrett, L., Thibaux, R., and Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440.

Petrov, S. and Klein, D. (2007). Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 404–411.

Petrov, S. and McDonald, R. (2012). Overview of the 2012 Shared Task on Parsing the Web. In *SANCL*, Montreal, Canada.

Seddah, D., Kübler, S., and Tsarfaty, R. (2014). Introducing the SPMRL 2014 shared task on parsing morphologically-rich languages. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages (SPMRL-SANCL)*, pages 103–109, Dublin, Ireland.

Seddah, D., Sagot, B., and Candito, M. (2012). The Alpage architecture at the SANCL 2012 shared task: robust pre-processing and lexical bridging for user-generated content parsing. In *SANCL 2012-First Workshop on Syntactic Analysis of Non-Canonical Language, an NAACL-HLT'12 workshop*.

Seddah, D., Tsarfaty, R., Kübler, S., Candito, M., Choi, J. D., Farkas, R., Foster, J., Goenaga, I., Gojenola Galletebeitia, K., Goldberg, Y., Green, S., Habash, N., Kuhlmann, M., Maier, W., Nivre, J., Przepiórkowski, A., Roth, R., Seeker, W., Versley, Y., Vincze, V., Woliński, M., Wróblewska, A., and de la Clergerie, E. V. (2013). Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, WA.

Sekine, S. and Collins, M. (1997). EVALB bracket scoring program. *URL: http://www. cs. nyu. edu/cs/projects/proteus/evalb*.

Wagner, J., Seddah, D., Foster, J., and Van Genabith, J. (2007). C-structures and F-structures for the British National Corpus. In *Proceedings of the Twelfth International Lexical Functional Grammar Conference*. CSLI Publications.

Domain Adaptation in Dependency Parsing via Transformation Based Error Driven Learning

Atreyee Mukherjee, Sandra Kübler

Indiana University, Bloomington, IN

atremukh@indiana.edu, skuebler@indiana.edu

ABSTRACT

Dependency parsers generally perform well when they are trained and tested on data from the same domain. However, if the data set on which we use the parser is different from the sentences on which it is trained, results tend to be low. Addressing this problem via domain adaptation is still a challenging problem and has achieved only limited improvements in the target domain. One problem that has been ignored to date concerns the differences in annotation schemes between corpora from different domains, even when the annotations are based on the same underlying annotation guidelines. In the most extreme case, the target annotations may contain labels that do not occur in the source domain. This significantly affects the overall performance of the parser. This paper presents an approach of applying transformation based error driven learning (TBL) for domain adaptation of dependency parsing. We use TBL to learn dependency label corrections in the target domain, based on errors made by the source domain parser. We show that this method can reduce dependency label errors significantly. A major advantage of this method is that we can address all types of errors with this method. The method can also be easily applied to any domain without any major change to the rule templates.

KEYWORDS: Domain adaptation; dependency parsing; transformation-based error-driven learning.

1 Introduction

Dependency parsing results tend to be reliable as long as the parser is trained and tested on data from a single domain. However, the situation is considerably more challenging when the data set on which the parser is tested/used is different from the sentences on which it is trained. For example, a parser trained on newspaper corpus would not parse texts based on spontaneous dialogues accurately. Since it is impossible to manually annotate the amount of data needed to train a parser in any domain that we need to parse, we need automatic methods to adapt a parser to those new domains. This problem is generally addressed in domain adaptation. Domain adaptation attempts to use a large set of annotated data from a different domain plus specific strategies to adapt the annotations to the target domain, for which a small amount of annotated data may exist. The problem is compounded when there are differences in the annotation schemes between the source and target domain. Different treebanks, representing different domains, tend to use somewhat different annotations (Dredze et al., 2007). These differences can be due to individual syntactic interpretations of different annotators, but in many cases, they are necessitated by phenomena in the target domain that do not exist in the source domain. Spontaneous dialogues, for example, have a large number of incomplete sentences with words that cannot be attached easily if their head is missing (e.g., the determiner in "I bought the -").

Out-of-domain dependency parsing results tend to be low as compared to in-domain results. The problem can be looked at from different perspectives. Previous work in this area has extensively looked at the structural errors, where the dependency arcs are corrected. For instance, the DeSR parser (Attardi et al., 2007; Attardi and Ciaramita, 2007; Attardi et al., 2009) implements tree revisions. Structural change of dependency trees prove to be an effective domain adaptation method. Yu et al. (2015) report a 1.6% improvement by using self-training to generate training examples for the target domain. To our knowledge, no methods have been reported for addressing functional errors, i.e., errors in the dependency labels.

Our work focuses on a target domain of spontaneous dialogues, using the CReST Corpus (Eberhard et al., 2010), which uses labels that do not occur in the source domain corpus. We propose to use transformation based error driven learning (TBL) (Brill, 1995) to address the problem, with a focus on correcting dependency labels. The method has been proven effective in a variety of NLP problems including POS tagging and syntactic parsing. The idea is simple: We use a small annotated data set in the target domain and automatically annotate it using a source domain parser. Our goal is to learn a set of rules from the errors that occur and a set of rule templates. Although we focus on correcting dependency labels in the current paper, our method can be extended to correct the errors in dependency arcs as well. We demonstrate that using TBL with a small target domain training set improves dependency parsing accuracy by about 10 % absolute, and it learns to use the target-specific labels.

The paper is structured as follows: Section 2 discusses related work, section 3 describes the issues in domain adaptation in more detail, and section 4 explains our approach of using TBL for domain adaptation. In section 5, we describe our experimental setup, and section 6 discusses the results. Section 7 concludes and delineates future work.

2 Related Work

There has been a significant amount of work done on domain adaptation in dependency parsing. The primary challenge of domain adaptation is the unavailability of annotated examples from

the target domain. Previous work in this area focused on analyzing how this affects parsing quality and on building systems to bypass the need for having a large amount of target domain data. Although distributional difference between the domains is a common problem, in general, Dredze et al. (2007) conclude that domain adaptation is most challenging when there are dissimilarities in annotation schemes between the domains.

Domain adaptation for dependency parsing was one of the tasks in the CoNLL 2007 Shared Task. The shared task was focused on the scenario where there is no data available from the target domain. Out of the 10 systems participating in the domain adaptation task, the highest results (81.06% LAS; 83.42% UAS) are achieved by Sagae and Tsujii (2007). To add target domain sentences to the training set, they emulate a single iteration of co-training by using MaxEnt and SVMs, selecting the sentences where both models agreed. The system by Attardi and Ciaramita (2007) (80.40% LAS; 83.08% UAS) produced similar results. They use a tree revision method (Attardi and Ciaramita, 2007) to correct structural mistakes. They formulate the problem as a supervised classification task using a multiclass perceptron, where the set of revision rules is the output space and features are based on syntactic and morphological properties of the dependency tree.

A popular approach for domain adaptation is selecting appropriate training data by using self-training or co-training for domain adaptation. Although testing on a single domain, McClosky et al. (2006a,b) show the effectiveness of using reranking and self training. They achieve an improvement of around 1% on unlabeled data. Kawahara and Uchimoto (2008) devised a method to select reliable parses from the output of a single dependency parser, MST parser (McDonald et al., 2005), instead of using an ensemble of parsers for domain adaptation. They use a self training method and combine labeled data from target domain with unlabeled data from the source by "concatenation". To estimate whether a parse is reliable, they apply binary classification using SVM on features such as sentence length, dependency length, unknown words, punctuation, average frequency of words in a sentence. They report a 1% increase in accuracy over the contemporary state of the art CoNLL shared task results on the shared task data. Yu et al. (2015) applied self-training for domain adaptation using confidence scores to select appropriate parse trees. Their highest improvement in terms of LAS is 1.6% on the CoNLL data.

Blitzer et al. (2006) used structural correspondence learning (SCL) for POS tagging and parsing to find "frequently occurring" pivot features, i.e., features that occur frequently in unlabeled data and equally characterize source and target domains. They used the WSJ as the source and MEDLINE abstracts as the target domain. They established that SCL reaches better results in both POS tagging and parsing than supervised and semi-supervised learning, even when there is no training data available on the target domain.

Transformation based error driven learning (TBL) was originally developed for POS tagging (Brill, 1992, 1995). Brill (1993) also used this approach to parse text by learning a "tranformational" grammar. The algorithm repeatedly compares the bracketed structure of the syntactic tree to the gold standard structure and learns the required transformations in the process. Brill and Resnik (1994) also use this technique for prepositional phrase attachment disambiguation. Transformation based error driven learning has also been used for information retrieval by Woodley and Geva (2005).

	Cases	# instances
1	Incorrect dependency label (incl. correct & incorrect heads)	13 839
2	Incorrect dependency head (incl. correct & incorrect label)	10 294
3	Incorrect dependency label & correct dependency head	5 389
4	Incorrect dependency label & incorrect dependency Head	8 450

Table 1: Analysis of sentences from CReST containing incorrect dependency predictions.

3 Issues in Domain Adaptation

Before we present our approach to domain adaptation, we need to better understand the different facets of the domain adaptation problem. This analysis will motivate our solution, which can handle all of those cases.

3.1 Error Types

The first phenomenon that we look at is the types of errors that can occur in dependency parsing:

1. Predicting the **head of a dependency arc** inaccurately, resulting in a structural error.
2. Predicting the **label of a dependency arc** wrong, resulting in a functional error.
3. Predicting both **label and the head of a dependency arc** wrong, resulting in structural and functional problems.

Note that structural errors affect the labeled and unlabeled attachment scores, functional errors only affect labeled attachment scores.

Previous work focused on correcting structural errors introduced by inaccurate prediction of the head of a dependency arc. To our knowledge, there are no approaches to address functional errors.

Next, we need to determine how serious these problems are. We concentrate on a setting where the source domain is the WSJ part of the Penn Treebank (Marcus et al., 1994) and the target domain is the CReST Corpus (Eberhard et al., 2010) (for more details on the corpora, see section 5). We now take a closer look at a parsing experiment where we use 17 181 WSJ sentences for training the MATE parser (Bohnet, 2010) and 5 759 CReST sentences for testing. The different types of errors are shown in table 1. This analysis shows that functional errors, i.e., errors involving dependency labels, are more frequent than structural errors. Thus, by ignoring those errors in domain adaptation, we artificially limit the possible improvements.

3.2 Error Sources

A second type of difference concerns the source of the parser errors. Here we need to distinguish two cases:

1. Parser inherent errors, i.e., errors that the parser makes independent of the out-of-domain setting.
2. Parser errors caused by differences in distribution between the two domains.
3. Differences in annotation, i.e., the domain data on which we evaluate differ from the annotations in the source data.

In the following, we focus on the differences of type 2 and 3. More specifically, we work on errors in the dependency labels since these are easier to distinguish. Structural differences tend

Treebank	No. of dep. labels	Intersection with WSJ
WSJ	67	-
CReST	31	22
Brown	71	60

Table 2: Differences in the dependency label sets across treebanks.

to be differences of degree, which are difficult to separate into distributional differences (type 2) and annotation differences (type 3). In order to establish whether annotation differences cause issues, we have analyzed a range of treebanks that are annotated using (variants of) the label set, i.e., the labels in the Penn Treebank when converted to dependencies using *pennconverter*. (Johansson and Nugues, 2007).

Table 2 looks at differences in the dependency label sets in the following treebanks: the WSJ part of the Penn Treebank, the Brown Corpus (Francis and Kucera, 1979) (Sections cg, ck, cl, cm, cn, cp, cr), and the CReST Corpus (Eberhard et al., 2010). The CoNLL style dependency trees of the WSJ and the Brown Corpus are created using the pennconverter tool (Johansson and Nugues, 2007); CReST was natively annotated in dependencies as well as constituents. It is obvious that there are considerable differences in the annotation schemes, especially between WSJ and CReST, which only have 22 labels in common. We can establish three different scenarios:

1. The source is a superset of the target data set.
2. The source is a subset of the target data set,
3. Both the source and target annotations have labels that do not occur in the other data set.

In the first case, the distribution of labels may be different between source and target, which can cause incorrect parsing decisions. The problem in the second case is more difficult since the parser is supposed to predict labels that it has not seen in training. The third case is a combination of the first two and thus the most difficult to handle.

Returning to our treebanks, we assume that since WSJ has the largest number of manually annotated sentences, it will serve as the source domain. In this case, CReST mainly shows a subset of labels from WSJ, but also has 9 labels that do not occur in WSJ.

Our hypothesis is that we can address *all* of the problems described above by using transformation based error driven learning (Brill, 1995) (for a description of this method see section 4). The method has been proven effective in many tasks such as POS tagging, syntactic parsing, and machine translation.

For the work presented here, we will focus on cases where the parser has predicted an incorrect label. This is motivated by our findings in table 1, which show that label errors are more frequent than structural ones. We investigate the scenario using CreST, where the source has a superset of labels in the target annotations, but with some unique labels in the target annotations. We hypothesize that our method can lead to an improved performance of the parser on the target domain, including dependency labels that do not exist in the source treebank. We evaluate this by the Labeled Attachment Scores (LAS) and Label Accuracy (LA).

4 Transformation-Based Error-Driven Learning for Domain Adaptation

4.1 TBL: The Brill Tagger

We implement the idea of transformation based error driven learning, which was originally developed by Brill (1992) for POS tagging. The method works with two versions of the training data: the gold standard and (an initially unannotated) working copy that simulates the learning process and records the errors. The unannotated version of the training data is tagged by a simple part of speech tagger[1], to create the initial state of the working copy of the text. The goal is to bridge the difference between the gold standard text and the working copy by learning rules that change the incorrect POS tags to the correct ones.

Learning is based on rule templates, which consist of two parts: rewrite rules and triggering environment. The templates need to be determined by the researcher, based on the problem. In general, a rule template is of the form:

Rule Template $(A, B) : X \rightarrow Y$

I.e., if we observe conditions A and B (triggering environment), we change the output variable (POS tags, for Brill and dependency labels, for us) from X to Y (rewrite rule). Note that X can remain unspecified, then the rule is applied independent of the current variable (label or POS tag).

The following is an example of a rule template for POS tagging: Change the POS tag of the current word from X to Y if the previous word is tagged as Z, where X, Y, and Z are variables that need to be instantiated during learning.

The learner creates rule hypotheses out of those templates by identifying incorrectly tagged words in the working copy and instantiating the template variables. For example, one possible rule hypothesis could be the following: Change the POS tag of the word “impact” from verb to noun if the previous word is tagged as a determiner.

In one pass through the system, all possible rule hypotheses are created and ranked based on an objective function which determines for each hypothesis how many corrections occur. The rule hypothesis with the highest score is applied to the working copy and added to the final list of transformations, stored in order, during the training process. Then the process repeats, and new rule hypotheses are generated from the templates based on the remaining errors in the working copy. The process finishes when there is no more improvement in terms of reduction of errors.

4.2 Domain Adaptation Using TBL

We use the underlying concept of transformation based error driven learning for reducing the dependency label errors resulting from parsing across domains. In particular, we address the scenario where the target domain contains a subset of the labels in the source annotation, but also contains new labels.

To use TBL for domain adaptation in dependency parsing, we need to adapt the following parts: In our case, the initial state consists of the training set from the target corpus parsed using a dependency parser, which was trained on the treebank from the source domain. We need to choose rule templates describing the relevant information that can help decide in which case to

[1] This could be any simple part of speech tagger, or a heuristic that labels every work with the most frequent POS tag

change a dependency label. We use POS and lemma[2] information rather than word forms in order to balance generality and specificity in the transformation rules. We utilize information on head of a word and its dependents since these provide essential contextual information. We currently do not use dependency labels as variables in our rule templates, but using this type of information is a viable modification in the future.

We use the following rule templates to describe the contextual properties of situations when a dependency label needs to be changed. Note that it is straightforward to include more templates, and in doing so, we can also model cases of structural change. In the latter case, the replacement part would take the form of "replace the current head of the word by word X". This will need to be accompanied by a check to ensure that the resulting dependency graph is still valid.

For this paper, we examine the following four rule templates.

- Rule template 1 (RT1)- $[(P_S, P_P, P_D, L_S, L_P, L_D) \rightarrow D_1]$

  If the part of speech tags of the word, its parent, and its dependent(s) are P_S, P_P and P_D, and the lemma of the word, its parent, and its dependent(s) are L_S, L_P, L_D respectively, the dependency label should be D_1. We consider all the dependents of a word as a single variable. E.g., $[(NN, VBZ, [DT, JJ, CC], box, be, [a, pink, and]) \rightarrow PRD]$

- Rule template 2 (RT2)- $[(P_S, P_P, L_S, L_P) \rightarrow D_1]$

  If the part of speech tags of the word and its parent are P_S and P_P, and the lemma of the word and its parent are L_S and L_P respectively, the dependency label should be D_1.

- Rule template 3 (RT3) - $[(P_S, P_P, P_D) \rightarrow D_1]$

  If the part of speech tags of the word, its parent and its dependent(s) are P_S, P_P and P_D respectively, the dependency label should be D_1.

- Rule template 4 (RT4) - $[(P_S, P_P) \rightarrow D_1]$

  If the part of speech tags of the word and its parent are P_S and P_P respectively, the dependency label should be D_1.

Figure 1 shows the TBL approach through four iterations of creating rule hypotheses from the templates, selecting and applying the best one.

We outline the process in algorithm 1. The learning iterates until there are no more rule hypotheses that would cause a further reduction in errors. During each iteration, based on the templates, the algorithm creates rule hypotheses from the errors found by comparing the working copy to the gold standard. It then chooses the best rule hypothesis that maximally reduces the errors in the working copy and applies this rule to the working copy. Since the templates that we use are independent of each other, we currently apply the top 10 rules per iteration. Choosing to apply multiple rules speeds up the overall process. To prevent over-generation, we favor the more specific hypothesis if there is a tie in the scores. We then store these rules in the final list of learned rules. After the application of the rules chosen in each iteration, we use the updated work copy in the next iteration.

After learning, a new text is processed by first having it parsed by the source domain dependency parser and then applying the learned rules in order.

[2]We derive lemma information using the TreeTagger (Schmid, 1994)

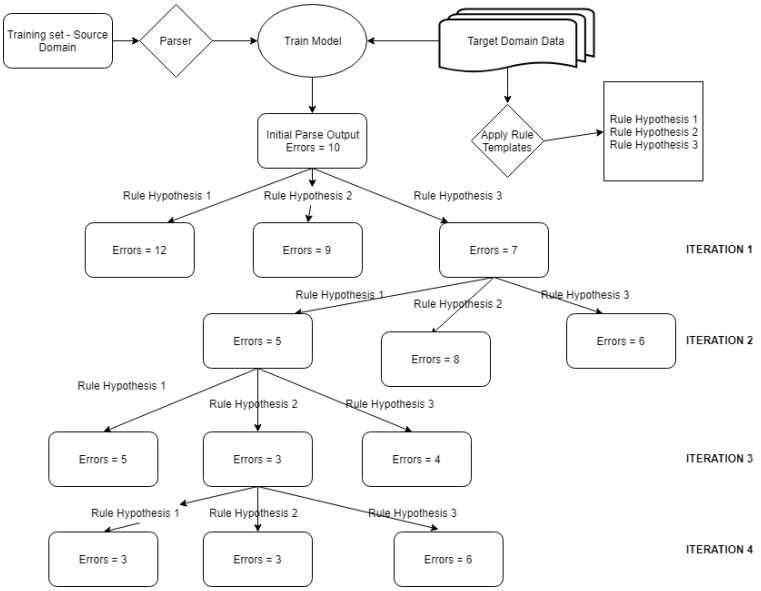


Figure 1: Reducing dependency label errors via transformation-based error-driven learning

Algorithm 1 TBL for Domain Adaptation

procedure *TBL_learn*(*training_data*)
    work_copy = *extract text from training_data; parse using WSJ model*
    learned_rules ← []
    repeat
        ▷ Generate rule hypothesis from errors in corpus given rule templates
        rule_hypotheses = *generate_hypotheses*(*training_data*, *work_copy*)
        ▷ calculate improvement score for each rule: # errors fixed
        rule_scores = *determine_scores*(*rule_hypotheses*, *work_copy*)
        ▷ Rank rules given scores; select rule with highest score
        best_rule, *best_score* = *argmax*(*rank_rules*(*rule_hypotheses*, *rule_scores*)
        ▷ If best rule causes improvement, apply rule to work copy
        if *best_score* > 0 **then**
            work_copy = *apply_rule*(*work_copy*, *best_rule*)
            learned_rules += *best_rule*

    until *best_score* <= 0
    return *learned_rules*

procedure *TBL_apply*(*learned_rules*, *test*)
    test_annotated = *parse test data using WSJ model*
    ▷ Apply learned rules in order to test set
    for each rule in learned_rules **do**
        test_annotated = *apply_rule*(*test_annotated*, *rule*)

WSJ	In an Oct. 19 review of " The Misanthrope " at Chicago 's Goodman Theatre (" Revitalized Classics Take the Stage in Windy City , " Leisure & Arts) , the role of Celimene , played by Kim Cattrall , was mistakenly attributed to Christina Haag .
CReST	it's like as soon - like if I were to close the door it's right next to like the bottom of the floor like where the door closes

Table 3: Sample sentences from Wall Street Journal (WSJ) & CReST corpus

	# Dialogues	# Sentences	
		with errors	without errors
Training Set	19	4384	459
Test Set	4	831	85

Table 4: Target domain training and test set for TBL

5 Experimental Setting

5.1 Data Sets

In our current work, we focus on two domains: financial news articles and dialogues in a collaborative task. We use the Wall Street Journal (Marcus et al., 1994) and the CReST corpus (Eberhard et al., 2010) as representative corpora for these two domains. For both corpora, we use teh dependency version: CReST was originally annotated in constituents and dependencies, the Penn Treebank was automatically converted to dependencies using *pennconverter* (Johansson and Nugues, 2007).

These corpora are very dissimilar in nature. On an average, the length of the sentences for WSJ is 24 and 7 for CReST. In terms of linguistic factors, CReST has a subset of WSJ's dependency labels with 10 new labels added. Example sentences from each corpora are shown in table 3. We use WSJ as the source domain and CReST as the target domain.

Target Domain The CReST corpus consists of 23 dialogues that were manually annotated for dependencies. We randomly select 19 dialogues as our training data for the TBL algorithm and the rest as test. Since the system needs to learn rule hypotheses from incorrect predictions of the source domain parser, we can safely ignore sentences that were parsed completely correctly. For the test data, we use all the sentences from the designated dialogues (with correct and incorrect dependency label predictions). Table 4 shows the division of sentences for training and test.

5.2 TBL Components

Initial State System We use the MATE parser (Bohnet, 2010) as the initial state annotator. We use 17 181 sentences from WSJ to train the MATE parser. We use this model to parse the sentences from the CReST training set to obtain the initial state annotated text.

Baseline We benchmark our results against the performance of the MATE parser trained on WSJ sentences, i.e., the out-of-domain parser, without any modification. We also report results for an in-domain setting where the MATE parser is trained on the CReST training set.

	Setting	# Incorrect Labels	LAS	LA	EM
1	Baseline (trained on WSJ)	2 112	59.02	63.73	8.41
2	In-domain baseline (trained on CReST)	572	**87.39**	90.18	**67.58**
3	Rule Templates (1+2)	600	69.91	89.70	48.03
4	Rule Templates (1+2+3+4)	451	***70.10***	**92.25**	***48.80***

Table 5: Results of applying TBL rules on test set.

Label	# in Gold	% Correct			
		baseline	in-domain	2 templates	4 templates
INTJ	690	0	97.39	98.26	99.71
ROOT*	195	0	67.35	63.59	77.44

Table 6: Accuracy in predicting CReST-specific labels.

5.3 Evaluation

For evaluation, we use the CoNLL-X script. We report the Labeled Attachment Scores (LAS) and Label Accuracy (LA) as the primary metrics for evaluating our system. LAS measures the percentage of words which have the correct head and dependency label. LA measures the number of correctly assigned labels. Reporting LA is important since this measure focuses solely on the accuracy of the labels while LAS also evaluates structural issues in the parse, which we do not address currently. We also report the exact syntactic match (EM), which measures the percentage of sentences that has correct overall predicted dependency trees.

6 Experimental Results

In this section, we discuss the outcome of our experiments for the TBL process. The results of applying the rules learned from TBL are given in table 5.

The first setting is our baseline. We evaluate the performance of our system against the results of parsing the sentences from the test set with the model trained on WSJ. The second baseline consists of an in-domain parser trained on the small CReST training set. Settings 3 and 4 evaluate the performance of the learned rules. The third setting derives rule hypotheses instantiated by 2 rule templates (rule templates 1 & 2). Since rule templates 1 & 2 are more specific (it accounts for lemma as well as part of speech tags), we add more general templates such as POS tags specifically (rule templates 3 & 4) to investigate if that leads to a significant difference in the results.

From table 5, we observe that applying TBL has a significant effect on the quality of the predicted dependency labels, with an improvement in LAS of almost 20% absolute over the out-of-domain parses. The number of completely correct sentences shows an improvement of 40% absolute over this baseline. We also see that extending the set of templates to include less-specific templates (templates 3 and 4) using POS tags information, has a minimal effect on LAS but increases label accuracy (LA) by 2.5% absolute.

When comparing the TBL results to the in-domain baseline, we notice that the baseline reaches an LAS considerably higher than the TBL results, but the TBL approach using all 4 templates reaches higher results with regard to LA (around 2% improvement). The different trends can be explained by the fact that we are currently restricting the TBL approach to functional changes, and we ignore structural errors. Since LAS evaluates both aspects, we see that the in-domain parser fares better in this respect. However, the LA, which only evaluates dependency labels,

Rule Template	Lemma (self)	Pos (self)	Lemma (head)	POS (head)	Lemma (children)	POS (children)	Deprel	Improvement
RT2	okay	UH	(root word)	(root word)	-	-	INTJ	1292
RT2	um	UH	(root word)	(root word)	-	-	INTJ	386
RT2	be	VBZ	(root word)	(root word)	-	-	ROOT	384
RT2	alright	UH	(root word)	(root word)	-	-	INTJ	216
RT2	yeah	UH	(root word)	(root word)	-	-	INTJ	205
RT2	the	DT	(root word)	(root word)	-	-	ROOT*	131
RT3	and	CC	be	VBZ	(no children)	(no children)	COORD	117
RT2	go	VBI	(root word)	(root word)	-	-	ROOT	100
RT3	that	DDT	be	VBZ	(no children)	(no children)	SBJ	83
RT2	well	UH	(root word)	(root word)	-	-	INTJ	77

Table 7: Top learned rules for the setting using 2 templates.

Rule Template	Lemma (self)	POS (self)	Lemma (head)	POS (head)	Lemma (children)	POS (children)	Deprel	Improvement
RT4	-	UH	-	(root word)	-	-	INTJ	2886
RT5	-	UH	-	(root word)	-	(no children)	INTJ	2766
RT2	okay	UH	(root word)	(root word)	-	-	INTJ	1292
RT3	okay	UH	(root word)	(root word)	(no children)	(no children)	INTJ	1271
RT4	-	AP	-	(root word)	-	-	INTJ	453
RT5	-	AP	-	(root word)	-	(no children)	INTJ	438
RT2	um	UH	(root word)	(root word)	-	-	INTJ	386
RT2	be	VBZ	(root word)	(root word)	-	-	ROOT	384
RT3	um	UH	(root word)	(root word)	(no children)	(no children)	INTJ	383
RT5		XY		(root word)		(no children)	ROOT*	268

Table 8: Top learned rules for the setting using 4 templates.

shows that our method is successful in improving the labels. Since the results of the experiment with 2 templates are similar to the in-domain treebank, we assume that we need to experiment with more specific templates in the future.

There are 9 dependency labels in CReST which do not occur in WSJ. Out of these, 2 labels (ROOT*, INTJ) occur in the test data. ROOT* is used for words whose head is missing because the sentence was interrupted; and INTJ is used for interjections. Since WSJ does not use these labels, these are predicted incorrectly in all the cases in the out-of-domain baseline.

We investigate the number of times these labels have been corrected by TBL. Table 6 shows the results. As expected, the labels do not show up in the baseline setting. For the settings using 2 or 4 templates, there is a significant improvement for the labels INTJ and ROOT*. TBL predicts these labels more accurately than the in-domain setting as well.

Tables 7 and 8 show the 10 learned rules with the highest scores, for the setting with 2 or 4 templates respectively. We can observe that many of the rules concern the labels INTJ and ROOT*. Since these labels do not appear in the source domain, applying rule hypotheses specific to these labels leads to the highest gains.

To have a closer look, we extracted the most frequent label confusions for each setting after parsing and TBL domain adaptation. The results are shown in table 9. For the baseline, the most frequently incorrect label is INTJ, followed by ROOT and ROOT*. By applying TBL, we have countered the most frequent label confusion as evident from the settings using 2 or 4 templates, where the numbers are lower across the board, but also the types of confusion are more typical of standard parsing issues (e.g., subject (SBJ) vs. predicate (PRD)).

7 Conclusion and Future Work

In this paper, we introduce transformation-based error-driven learning (TBL) for domain adaptation in dependency parsing. Since there tend to be significant differences between annotation

Baseline			2 Templates			4 Templates		
Gold	Predicted	Counts	Gold	Predicted	Counts	Gold	Predicted	Counts
INTJ	ROOT	354	ROOT*	ROOT	39	ROOT*	ROOT	38
INTJ	DEP	210	SBJ	PRD	20	SBJ	PRD	25
ROOT	NMOD	136	ROOT*	NMOD	20	ROOT	ROOT*	21
ROOT*	NMOD	100	LOC	NMOD	19	DIR	LOC	19
INTJ	NMOD	85	ROOT	ROOT*	17	DIR	ADV	17
LOC	NMOD	55	DIR	ADV	15	LOC	NMOD	15
COORD	DEP	38	ROOT	NMOD	13	TMP	ADV	10
ROOT	COORD	37	DIR	NMOD	12	LOC	ADV	10
LOC	LOC-PRD	35	LOC	ADV	11	ROOT	NMOD	8
LOC	PRD	33	PMOD	NMOD	10	OBJ	SBJ	8

Table 9: The 10 most frequent label confusions across the different settings.

schemes of different corpora from different domains, we focus on methods that can address those differences systematically along with differences due to the domain itself. When a text is parsed with a model trained on one domain, it leads to a significant number of incorrect predictions, especially for dependency labels. We address this problem in our work by using TBL to learn rules from a small target domain training set and a small set of manually defined rule templates. In comparison to a pure out-of-domain parser, we observe a significant increase in the labeled attachment scores (∼20%), labeled accuracy (∼30%) and exact match (∼40%) for WSJ as source and CReST as target corpus. The observed improvement is largely due to the labels that are used in CReST, but do not occur in WSJ since those are mislabeled consistently by the out-of-domain parser. However, when we apply TBL, these labels are corrected in most of the cases. When we compare our results to an in-domain parser, the results show that the TBL approach is very good at correcting the dependency labels, but we need to extend the approach and use more specific templates and cover structural errors as well to be able to compete with the in-domain parser with regard to LAS.

Our method is flexible in that any number of variables can be used in the templates, and it can be extended to cover structural changes, i.e., to correct dependency head annotations. In addition, the templates are valid across different target domains.

As a part of our future work, we plan to evaluate the effectiveness of this method for multiple domains. We will also introduce new variables in our rule templates including contextual information on dependency labels. In addition to correcting dependency labels, we will extend the method to correct predicted dependency heads by introducing new templates. This will need to be accompanied by a check to ensure that the resulting analysis is still a valid dependency graph.

References

Attardi, G. and Ciaramita, M. (2007). Tree revision learning for dependency 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 388–395.

Attardi, G., Dell'Orletta, F., Simi, M., Chanev, A., and Ciaramita, M. (2007). Multilingual dependency parsing and domain adaptation using DeSR. In *EMNLP-CoNLL*, pages 1112–1118.

Attardi, G., Dell'Orletta, F., Simi, M., and Turian, J. (2009). Accurate dependency parsing with a stacked multilayer perceptron. *Proceedings of EVALITA*, 9:1–8.

Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 120–128, Sydney, Australia.

Bohnet, B. (2010). Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 89–97, Beijing, China.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 112–116, Harriman, NY.

Brill, E. (1993). Automatic grammar induction and parsing free text: A transformation-based approach. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 259–265.

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 24(1):543–565.

Brill, E. and Resnik, P. (1994). A rule-based approach to prepositional phrase attachment disambiguation. In *Proceedings of the 15th Conference on Computational Linguistics*, pages 1198–1204.

Dredze, M., Blitzer, J., Talukdar, P. P., Ganchev, K., Graca, J., and Pereira, F. C. (2007). Frustratingly hard domain adaptation for dependency parsing. In *EMNLP-CoNLL*, pages 1051–1055.

Eberhard, K., Nicholson, H., Kübler, S., Gunderson, S., and Scheutz, M. (2010). The Indiana "Cooperative Remote Search Task" (CReST) Corpus. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, Valetta, Malta.

Francis, W. N. and Kucera, H. (1979). Brown corpus manual. Brown University.

Johansson, R. and Nugues, P. (2007). Extended constituent-to-dependency conversion for English. In *Proceedings of NODALIDA 2007*, pages 105–112, Tartu, Estonia.

Kawahara, D. and Uchimoto, K. (2008). Learning reliability of parses for domain adaptation of dependency parsing. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP)*, Hyderabad, India.

Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., and Schasberger, B. (1994). The Penn Treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 114–119, Plainsboro, NJ.

McClosky, D., Charniak, E., and Johnson, M. (2006a). Effective self-training for parsing. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 152–159, New York, New York.

McClosky, D., Charniak, E., and Johnson, M. (2006b). Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 337–344, Sydney, Australia.

McDonald, R., Pereira, F., Ribarov, K., and Hajič, J. (2005). Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530.

Sagae, K. and Tsujii, J. (2007). Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1044–1050, Prague, Czech Republic.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*.

Woodley, A. and Geva, S. (2005). Applying transformation-based error-driven learning to structured natural language queries. In *International Conference on Cyberworlds*, pages 8–pp.

Yu, J., Elkaref, M., and Bohnet, B. (2015). Domain adaptation for dependency parsing via self-training. In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 1–10.

Seq2Seq or Perceptrons for robust Lemmatization. An empirical examination.

Tobias Pütz, Daniël De Kok, Sebastian Pütz, Erhard Hinrichs

SFB833 A3, University of Tübingen

{tobias.puetz, daniel.de-kok, erhard.hinrichs}@uni-tuebingen.de

sebastian.puetz@student.uni-tuebingen.de,

ABSTRACT

We propose a morphologically-informed neural Sequence to Sequence (Seq2Seq) architecture for lemmatization. We evaluate the architecture on German and compare it to a log-linear state-of-the-art lemmatizer based on edit trees. We provide a type-based evaluation with an emphasis on robustness against noisy input and uncover irregularities in the training data. We find that our Seq2Seq variant achieves state-of-the-art performance and provide insight in advantages and disadvantages of the approach. Specifically, we find that the log-linear model has an advantage when dealing with misspelled words, whereas the Seq2Seq model generalizes better to unknown words.

KEYWORDS: Lemmatization, German, Error Analysis, Sequence2Sequence.

1 Introduction

Lemmatization is the process of mapping a form to its dictionary entry. Lemmas are a requirement to associate any inflected form with a lexical resource. In Natural Language Processing, lemmas are common features for a wide range of tasks and have been shown to improve results in parsing (Dozat and Manning, 2018) and machine translation (Sennrich and Haddow, 2016).

Besides being useful as features for statistical classifiers, lemmas are also of importance for other areas of linguistics. A common task in distributional semantics, for instance, is to algorithmically obtain a vector representing a certain word. A simple approach, which obtains discrete representations, computes the pointwise mutual information (PMI) between a word and its cooccurents. Word embeddings, in contrast to PMI, are continuous. They can be obtained through various methods, such as maximizing the similarity between word and context vectors (Mikolov et al., 2013). Both approaches are known to produce bad representations for rare words. This problem is especially relevant for morphologically-rich languages where the occurrences of rare words are divided between their possibly numerous different inflections. Building the word representations based on lemmas leads to less sparse representations, as the inflections of a word are seen as the same symbol, combining their occurrences. The sparsity issue also applies when querying a treebank of a morphologically-rich language. Here, a researcher might be interested in the usage of a certain word. Without the lemma as the common feature, every inflection of a word needs to be spelled out to obtain all usages. As a consequence, most treebanks contain lemmas as an annotation layer. As manual annotation is expensive and new methods require more data, we observe the rise of big web-corpora with automated annotation, which subsequently leads to a growing need for performant and robust lemmatization, fit for noisy web text.

In this work, we propose a morphologically-informed variant of the recently successful Sequence to Sequence (Seq2Seq) architecture (Sutskever et al., 2014) for lemmatization (*Oh-Morph*) and provide an in-depth comparison with the Lemming system (Müller et al., 2015) on two German treebanks: TüBa-D/Z (Telljohann et al., 2004) and NoSta-D (Dipper et al., 2013). In contrast to other recent work, we train and evaluate on types such that there is no overlap between training and test set. By type we denote unique combinations of form, lemma, POS and morphological tags. Table 1 specifies the input and expected output of the lemmatizer:

Input			**Output**
Form	*POS*	*Morphological tags*	*Lemma*
folgenden	NN	case:dative|number:singular|gender:neuter	folgendes
folgenden	ADJA	case:genitive|number:plural|gender:feminine	folgend

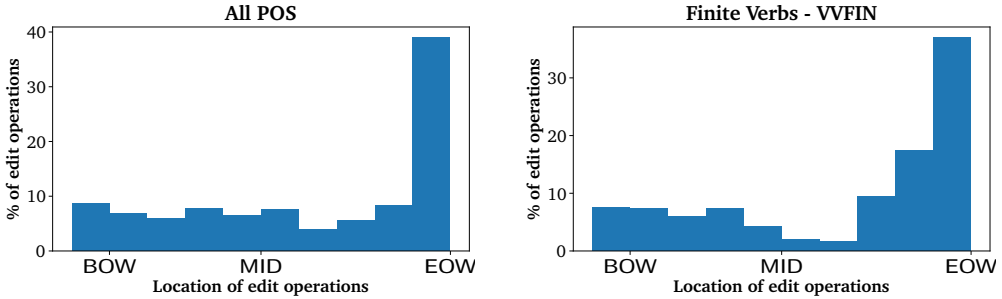
Table 1: Example of input and output of the lemmatizer.

The POS and morphological tags incorporate the necessary context information to lemmatize ambiguous forms. The type-based evaluation gives us the opportunity to perform a common model comparison but also to highlight problematic edge cases that would have been obscured under a token-based evaluation. To further our insight into the robustness of the models against noisy input, we use automatic morphological annotations instead of a gold standard both at training and test time. Moreover, we pay close attention to how the models deal with misspelled and unknown words. Given the usually well-formed newspaper texts in most treebanks, this is an aspect of the evaluation that is often overlooked.

We find that the log-linear *Lemming* outperforms *Oh-Morph* by a slight margin. *Lemming*, being able to leverage a word list, works best on lemmatizing non-standard language, like dialect

variants or misspelled words. It should be noted, though, that non-standard spelling is still one of the biggest error sources. *Oh-Morph*, in contrast, generalizes better to unknown words as we find *Lemming's* performance to deteriorate on out-of-vocabulary items. We conclude that both systems have their advantages and believe that further improvements can be made by improving their robustness against non-standard spelling of input words.

2 German Morphology



(a) macro-averaged edit operations

(b) finite verb edit operations

Figure 1: Locations of edit operations to transform forms to their lemma in TüBa-D/Z.

Inflectional patterns. As shown in Figure 1, inflections in German are largely limited to the suffix. However, certain verb forms deviate from this pattern and introduce prefixes as in *ge-schlossen* 'closed', the past participle of *schließen* 'close'. Another important exception are separable verbs, like *abschließen*. Such verbs have a prefix (e.g. *ab-* in *abschließen*) that appears separately when the verb is used as a finite verb in a declarative main clause. In such clauses, the verb is in the left bracket and the separable prefix appears in the right bracket, as demonstrated in (1).

(1) Als er erfuhr, dass er die Tür **abschließen** soll, **schloss** er sie **ab**.
When he heard, that he the door lock should, locked he it (prefix-ab).
'When he heard that the door should be locked, he locked it.'

Separable verbs have special infinitives and participles. They introduce the infix *-zu-*, a German infinitive marker, as in *ab-zu-schließen*, or *-ge-*, a participle marker, as in *ab-ge-schlossen*. Forms like *schließen*, where inflections change the parts of the root, are considered to be irregular. Irregular inflections are not limited to verbs but also occur, less frequently in other word classes such as adjectives, e.g. *gut* 'good', *besser* 'better' and *am besten* 'best'.

Challenges. A question that arises when dealing with separable verbs is whether their prefixes should be considered part of the lemma. Given the difference in meaning between e.g. *aufgeben* 'to give up' and *geben* 'to give' it would be very problematic to drop them. Keeping them, on the other hand, introduces the need to reattach separated prefixes which requires topological field or dependency annotations, steps usually performed after lemmatizing. A possible way out is to disregard the prefixes and reattach them once the required syntactical annotations have been made. This path, however, leads to the problem of deciding which of the possibly multiple prefixes is separable. Some can always be separated, some never, for others both separable and non-separable verb forms exist. In TüBa-D/Z (Telljohann et al., 2004),

lemmas mark separable prefixes with a '#' between prefix and stem, lemmatization of these forms then means to infix '#' for non-separated separables and to reattatch the prefix with the marker for separated ones. In the version used for this work, the prefixes have been removed in order to reach an homogeneity of annotation between both separated and non-separated separable verbs.

An additional challenge in the form of syncretism can be found in animate nouns. Some nouns with the masculine singular ending *-er* and the feminine *-in*, like *Schauspieler/-in* 'actor / actress' have no marked nominative plural for the masculine form. Others, like *Vorsitzenden* in (2), do not mark gender in nominative plural.

(2) Die Vorsitzenden trafen sich zum Krisengespräch.
The chair(wo)men met (refl) for-a crisis-meeting.
'The chairpersons met for a crisis meeting.'

Müller et al. (2015) mention that is important to know the lemma of these forms in order to assign a gender. We agree, however, in some cases the singular form can only be recognized if, possibly extra-sentential, discourse information is available. In other cases, e.g. if a plural word describes a mixed gendered group, the word cannot be reduced to a singular form since no un-gendered singular exists. As German grammar enforces gender, case and number congruency, syncretic forms are often disambiguated by accompanying determiners. While these prove to be useful in some cases, it should be noted that they display quite some ambiguity as well.

3 Background and Related Work

In the following section, we will first discuss existing lemmatization systems and then introduce methods relevant to our proposed model.

Previous work. Some early lemmatization systems employ finite state technology and solve morphological analysis and lemmatization as one task (Minnen et al., 2001; Schmid et al., 2004; Sennrich and Kunz, 2014). Given enough expert effort, these are able to achieve very good coverage. However, as their performance directly correlates with the completeness of their lexica, most transducers handle out-of-vocabulary items poorly. Moreover, as non-statistical tools they are not able to disambiguate syncretic forms. Others enrich the input with linguistic annotations and choose the correct transformation to the according lemma Chrupała (2006). Later a synergy between assigning these annotations and the lemma jointly was found (Chrupała et al., 2008; Müller et al., 2015). All these aforementioned approaches rely on sometimes language-specific sets of handcrafted features.

Lemming. The Lemming system (Müller et al., 2015), in the same vein as Chrupała (2006) and the Morfette system (Chrupała et al., 2008), treats lemmatization as a classification problem. During training, they derive edit-trees to transform a form into its lemma and then learn to choose the correct lemma from a set of candidates, generated by applying all possible edit-trees to a form. This simple approach achieves state-of-the-art performance on multiple languages. It greatly benefits from its ability to incorporate arbitrary global features, such as frequency counts or a word list. Müller et al. (2015) report improved performance by training Lemming and the Conditional Random Field (CRF) morphological tagger MarMot (Müller et al., 2013) jointly. However, their evaluation is bound to the token level, which we suspect to bias their evaluation towards frequent tokens, that we also expect to appear both in training and validation set.

Sequence to Sequence. The Sequence to Sequence (Seq2Seq) architecture (Sutskever et al., 2014) is a special variant of Recurrent Neural Networks (RNN). In contrast to regular RNNs, where the number of outputs is fixed, Seq2Seq enables mapping an arbitrary amount of inputs to an arbitrary number of outputs. These domain-agnostic models can be seen as feature-less and have achieved impressive results in several sequence transduction tasks, including machine translation (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2014), text summarization (See et al., 2017), constituency parsing (Vinyals et al., 2015), and closely related to lemmatization, morphological re-inflection (Cotterell et al., 2017, 2016). The common Seq2Seq architecture, also known as *encoder-decoder* network, consists of two RNNs. The encoder processes each input symbol in sequence while maintaining an internal state. The decoder is then initialized with the internal state of the encoder and predicts one symbol per step until a special end-of-sequence token is predicted. The input at every decoding step is the previously predicted token, with the first step receiving a special beginning-of-sequence token.

Attention. As the standard encoder-decoder architecture compresses its inputs into a fixed size vector, it struggles with long input sequences (Cho et al., 2014). A way to view this problem is that due to the longer input sequence, the encoder has to compress more information over a longer distance into the same dimensionality. Bahdanau et al. (2014) solve this issue by allowing the decoder to not only access the final state of the encoder but also each intermediate state, which they achieve through alignment mechanisms. Luong et al. (2015) simplified the alignment calculations by introducing the dot-product as scoring function for the attention mechanism. As both attention variants need to calculate the alignment weights for all encoder states at each decoder step, they have quadratic time complexity in $O(TU)$ where T and U are the lengths of the input and output sequence (Raffel et al., 2017). Raffel et al. (2017) reduce this to linear time with their monotonic Attention. It enforces linear alignments, where the decoder can only move forward in focusing on encoder states.

Seq2Seq lemmatization. Bergmanis and Goldwater (2018) applied the Seq2Seq architecture to lemmatization. They describe their approach as context sensitive, as the encoder processes not only the word form but also 20 characters of left and right context. In contrast to other systems, they do not require morphological or POS tags. However, as Müller et al. (2015), they evaluate on the token level. Schnober et al. (2016) compared pruned CRFs with Seq2Seq architectures and also evaluate on lemmatizing Finnish and German verbs taken from the Wiktionary Dataset (Durrett and DeNero, 2013). Besides limiting the task to verbs, they also lack a qualitative and exhaustive evaluation on the specific task of lemmatization.

4 Setup

In the following section, we will first describe the two variants of Lemming (Müller et al., 2015) that we used for comparison (*Lemming-Base*, *Lemming-List*), and then introduce our proposed model, *Ohnomore-Seq2Seq*.

4.1 Lemming

We use two variants of Lemming (Müller et al., 2015): *Lemming-Base* and *Lemming-List*. *Lemming-Base* utilizes its built-in features, including several alignment, edit-tree and lexical features. As Lemming supports the addition of arbitrary features, we also use *Lemming-List* in our experiments which adds a word list.[1] Both *Lemming-Base* and *Lemming-List* were trained using

[1]Available at `https://sourceforge.net/projects/germandict` accessed on 09.29.2018

the perceptron classifier with hashed features and morphological tags as additional features.

4.2 Ohnomore-Seq2Seq

Model. *Ohnomore-Seq2Seq* (*Oh-Morph*) (Pütz, 2018) closely resembles the classical encoder-decoder Seq2Seq architecture (Sutskever et al., 2014), extended with Luong-style monotonic Attention (Luong et al., 2015; Raffel et al., 2017). We dropped the reversing of the input, as we could not observe any differences in performance, likely due to attention which relaxes long-range dependencies. Furthermore, we concatenate the embedded morphological and POS tags with the final state of the encoder, resulting in a $d + p + (m * n)$ dimensional vector, where d is the state size of the encoder, p and m the size of morph- and POS- embeddings and n the maximal number of morphological tags encountered during training. This vector is then fed through a feed-forward layer with the SELU activation function (Klambauer et al., 2017), resulting in a vector with the dimensionality of the decoder's state size, which is the initial state of the decoder. Alternative setups, including bi-directional encoder, beam-search and a CRF-layer did not lead to improvements. It should be noted that the inclusion of a word list, as for *Lemming-List*, is not easily done, since *Oh-Morph* does not generate a candidate set ahead of time and therefore cannot include features of the lemma.

Hyperparameters. For training, we use mini-batches of 2,048 examples and discard forms longer than 60 characters. We use the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.03 and clip gradients with a norm bigger than 5. The character embeddings have 100 dimensions, POS embeddings 50 and morph embeddings 30. We apply a dropout of 0.8 on the input embeddings. As recurrent cells in the encoder and decoder we use LSTMs (Hochreiter and Schmidhuber, 1997) with a recurrent dropout of 0.5. We train for 10,000 steps and then stop after 15 epochs without an improvement.

5 Evaluation and Data

Corpus	# Tokens	# Types
TüBa-D/Z	1.8M	213,705
NoSta-D	39,504	5,643
NoSta-D w/o. TüBa-D/Z	34,504	4,010

Table 2: TüBa-D/Z, NoSta-D and NoSta-D without the TüBa-D/Z sub-corpus with token and type count.

Evaluation. In contrast to other recent work in lemmatization like Müller et al. (2015) or Bergmanis and Goldwater (2018), we decided to evaluate on types instead of tokens. We did so because we suspect that token-based evaluation is biased towards getting frequent tokens right. Moreover, it is to be expected that a token which ends up both in the training and validation set will be predicted right, simplifying the task. We perform 10-fold cross validation on our in-domain data and average the accuracy of the 10 models on the out-of-domain data.

Data. Table 2 reports token and type counts on our data sets: TüBa-D/Z (Telljohann et al., 2004), a treebank containing articles of the German newspaper Taz, and as out-of-domain data NoSta-D (Dipper et al., 2013), a corpus containing non-standard variations of German. To account for a realistic setting with potentially erroneous morphological tags, we used the state-of-the-art CRF morphological tagger MarMot (Müller et al., 2013) to annotate TüBa-D/Z and NoSta-D using 5-fold jackknifing. After tagging we filter duplicates, such that every combination of

form, *lemma*, *POS* and *morphological tags* is unique. We further remove irregular forms and closed class words,[2] as we consider it as impossible to infer the lemma when using on type level disjoint sets. Moreover, we suspect that both irregular forms and closed class words can be easily lemmatized using dictionaries. We retrieved a list with 2,039 irregular forms from Celex German (Baayen et al., 1993). Since NoSta-D's lemma column is also used for normalization on sentence level e.g. insertions of elided tokens, we filter all tokens where an appended '|' marks a continuation or where an empty form is mapped to a lemma.

6 Results and Discussion

	TüBa-D/Z	**NoSta-D**	**NoSta-D w/o. TüBa-D/Z**
Oh-Morph	97.00%	83.69%	79.41%
Lemming-Base	96.78%	83.45%	79.00%
Lemming-List	**97.02%**	**83.96%**	**79.73%**

Table 3: Accuracy on TüBa-D/Z, NoSta-D and NoSta-D without the TüBa-D/Z sub-corpus. *Lemming-List* outperforms *Oh-Morph* by a slight margin on all sets. *Lemming-Base* consistently performs the worst. Best results are bold.

	Oh-Morph	**Lemming-Base**	**Lemming-List**	**Shared**
# total	6,078	6,078	6,078	207,627
# errors	3,088	3,565	3,051	3,326
% errors	50.80%	58.65%	50.20%	1.60%

Table 4: Unique and shared predictions on TüBa-D/Z with error rates. There are 207,627 types where all models had identical predictions and 6,078 where one had a unique prediction. The error rate within the identical predictions is only 1.6%. The unique predictions have consistent error rates of more than 50%.

	TüBa-D/Z	**Falko**	**BeMaTaC**	**Anselm**	**Unicum**	**Kafka**
Oh-Morph	94.22%	89.83%	78.86%	27.19%	**75.38%**	91.07%
Lemming-Base	94.32%	90.16%	78.80%	26.44%	74.07%	90.91%
Lemming-List	**94.37%**	**90.93%**	**79.44%**	**30.33%**	74.56%	**91.08%**

Table 5: Accuracy on the different NoSta-D sub-corpora. *Lemming-List* shows the best performance across all sub-corpora apart from Unicum (online chats), here *Oh-Morph* achieves the highest accuracy. TüBa-D/Z: news paper texts, Falko: L2 learner language, BeMaTac: spoken language, Anselm: historical text, Unicum: online chats, Kafka: literary prose. Best results are bold.

Results. Table 3 provides the results on TüBa-D/Z and NoSta-D, the results on each NoSta-D sub-corpus will be discussed in the following section. We observe that *Lemming-List* shows the overall best performance. *Oh-Morph* performs slightly worse on TüBa-D/Z and by a bigger margin on NoSta-D. *Lemming-Base* shows the lowest performance across all sets. Table 4 dissects the results on TüBa-D/Z into two sets, shared and unique predictions, and presents the error rates on the respective set. The shared set contains the 207,627 types for which all three models produced the same output. The unique set consists of the 6,078 types for which at least one of

[2]Available at `http://www.sfs.uni-tuebingen.de/Elwis/stts/Wortlisten/WortFormen.html` accessed on 09.29.2018

the models produced an unique prediction. There is an approximate 50-50 split between the 3,051 to 3,565 model-unique and the 3,326 shared errors. The large number of shared errors might hint at issues within the training data like tagging errors.

NoSta-D sub-corpora. NoSta-D consists of six diverse sub-corpora, ranging from online chats over learner language to historic texts. Table 5 reports the accuracy on each subcorporpus. We find that both Lemming models show a better performance on L2 learner language. *Oh-Morph* seems to operate well on online chats whereas, *Lemming-List* shows the best results with spoken language and by a clear margin on historic German text. The overall performance on the historic text is bad, most likely due to spelling variations that are not common anymore and very far from their canonical spelling, like the form *vrouwe* with the standard spelling *Frau* 'woman'.

6.1 Error Analysis

In the following section we will work out model specific strengths and provide some insight in anormalities in the training data.

6.1.1 TüBa-D/Z

Analysis. We sampled 22,007 types from TüBa-D/Z and classified the incorrect portions of the predictions of the three models into 7 error classes described in the following section. Types for which all models made the same predictions will be discussed separately. For 600 at least one model had a unique prediction. For 21,407 the predictions were identical.

Error classes. For our analysis we assign the following 7 error classes:

1. **Unsolvable:** cases that fail due to annotation errors within the lemma, like spelling mistakes; and cases where sentential information is needed, e.g. a truncated form that is part of an enumeration that receives a full lemma.
2. **Spelling:** misspelled forms which the lemmatizer did not correct.
3. **NE:** errors connected to named entities.
4. **Separable:** verbs with a separable prefix where the model retained the prefix or other verbs where a prefix was mistakenly cut off.
5. **Wr. morph** cases where the predicted lemma corresponds to wrong morphological tags, e.g. a plural noun which was tagged as nominative singular where the inflected form was returned as the lemma.
6. **Ign. morph:** cases where correct morphological tags have been ignored, e.g. a form was changed where the morphological tags imply that the form is the lemma.
7. **Solvable** all cases we consider solvable that are not captured by the other classes. For example, predictions where a superfluous character remains as a suffix.

Apart from assigning these fine-grained classes, we also group the errors by whether we consider them solvable or not. The unsolvable group is already described by the **Unsolvable** class. As borderline solvable we consider:

- **Spelling** and **NE**, as they might require world knowledge;
- **Separable**, since we believe that these are hard to pick up when training on types; and
- **Wr. morph**, as correct morphological tags are a requirement to lemmatize syncretic forms.

The solvable group counts the two members:

- **Ign. morph** as the necessary information is present; and

- the **Solvable** class.

	Correct	Solvable	Ign. morph	Wr. morph	Separable	NE	Spelling	Unsolvable
Oh-Morph	49.50%	19.17%	4.33%	5.83%	2.00%	7.67%	9.33%	2.17%
Lemming-Base	42.50%	21.33%	6.00%	8.33%	5.83%	5.67%	8.17%	2.16%
Lemming-List	50.33%	18.50%	6.17%	7.50%	5.17%	5.67%	3.50%	3.16%

Table 6: Result of the analysis of 600 of the non-identical sampled predictions from *Oh-Morph*, *Lemming-Base* and *Lemming-List*. Spelling is the biggest single error cause for both list-less models. *Lemming-List* shows the most issues with morphological tags. **Correct** corresponds to the error rates presented in Table 4. Abbreviations: *wr.*: wrong, *ign.*: ignored, *NE*: named entity.

Result. The results of our analysis are presented in Table 6. We see that *Lemming-Base* and *Lemming-List* show more problems connected to morphological tags. *Oh-Morph* might benefit from its capability to form a fine-grained character-based morphological representation in addition to the morphological tags, enabling a decision whether a form-tag combination is valid or not. Linear classifiers, like the perceptron used by both Lemming variants, in contrast, cannot capture these feature interactions. Further, we find that *Oh-Morph* outperforms both *Lemming-List* and *Lemming-Base* on separable verbs. With named entities *Oh-Morph* displays issues, we especially find problems with word final *-s*. The name of the sports brand *Adidas*, for example, got reduced to *Adida*, as the *-s* was recognized as a genitive marker. These errors seem to stem from an uncertainty whether a genitive ending in *-s* is syncretic with the nominative or inflectional. With spelling errors *Lemming-List* shows the least errors. Given the clear margin between it and *Lemming-Base* we believe that the word list provides crucial information whether a generated candidate lemma is well-formed or not.

Intersection. In 3,326 cases all three models made identical errors. The analysis of 340 errors is provided in Table 7. We find that the biggest cause of shared errors is an inability to correct spelling errors (37.65%). Further outstanding are erroneous morphological tags (14.41%) and errors connected to named entities (13.82%).

	Solvable	Ign. Morph	Wr. morph	Separable	NE	Spelling	Unsolvable
Intersection	18.53%	6.47%	14.41%	2.35%	13.82%	37.65%	6.76%

Table 7: Result of the analysis of 340 errors of the 21,407 sampled identical predictions. Spelling is the biggest single cause of errors.

Vocab	Type	Oh-Morph	Lemming-Base	Lemming-List	% unknown
Train	Form	**95.74%**	95.21%	95.62%	48.97%
	Lemma	**96.32%**	96.04%	95.98%	34.09%
List	Form	**94.34%**	94.27%	94.20%	28.34%
	Lemma	**96.48%**	96.47%	95.70%	28.98%

Table 8: Average share of unknown lemmas and forms per validation-fold of TüBa-D/Z with accuracies of *Oh-Morph*, *Lemming-Base* and *Lemming-List*. *Oh-Morph* performs the best on all out-of-vocabulary items. *Train* rows report the accuracy on forms and lemmas not contained in the training data, *List* on items not contained in the word list of *Lemming-List*. Best results are bold.

Out of vocabulary. Table 8 quantifies the performance of the models on unknown forms and

lemmas.[3] We inspect how the models deal with out-of-vocabulary items with respect to two vocabularies: *Lemming-List's* word list and the respective training fold. It should be noted that neither *Lemming-Base* nor *Oh-Morph* use the word list, we include their results for the sake of comparison. For all vocabularies *Oh-Morph* seems to be suited best to deal with unknown items. *Lemming-List* performs surprisingly poorly on unknown lemmas and shows worse results than *Lemming-Base* on these items. It seems that while *Lemming-List* is able utilize its word list to deal with spelling errors, it also relies on its completeness and shows a drop in performance on unknown entities.

	Oh-Morph	**Lemming-Base**	**Lemming-List**	**Shared**
# total	171	171	171	283
% errors	85.07%	72.51%	71.93%	68.90%

Table 9: Unique and shared errors on dialect and colloquial language. None of the models is suited to deal with dialect variants.

Colloquial language. In TüBa-D/Z words from dialects and colloquial language with non-standard spelling are mapped to their standard spelling with a trailing underscore. These form-lemma pairs are often only vaguely related, sometimes through phonetic similarities as in *verschandölln-verschandeln_* 'to vandalize' or *Frollein-Fräulein_* 'miss', in other cases like *koscht-kosten_* 'to cost' they are contractions. A problem when dealing with these cases is that the borders between lemmatization and text normalization with spelling errors and dialectical variants become blurry. Some dialect forms can easily be mistaken for spelling errors, with others context might be needed to retrieve the canonical form. In total there are 454 types where a lemma has a trailing underscore. The error rates on these types are reported in Table 9. We find that none of the models is suited to deal with these types as none manages to predict the right word in more than 30% of the cases. Moreover, both Lemming models infer the right lemma twice as often as *Oh-Morph*.

Ambiguity. During the annotation process, we noticed several *form-pos-morph* combinations that were associated with more than one lemma. Further examination revealed that there were in fact 921 cases in which the training data contains contradictory examples. The vast majority of these are nouns and named entities, accounting for 803 cases. Both Lemming models have an error rate of 72% on these cases, while *Oh-Morph* fails in 66% of the cases to give the expected lemma. Most of the ambiguous examples are nominalized verbs like *(der) Gehende* 'the walker' / *(ein) Gehender* 'a walker' where for definite and indefinite a separate nominative singular exists. According to the TüBa-D/Z annotation guidelines (Telljohann et al., 2006) these forms should be lemmatized to the indefinite nominative singular. As our data is machine tagged we searched gold-standard TüBa-D/Z for ambiguous combinations and find 738 cases that cannot be lemmatized using our featurization. It remains to be explored whether some of these are disambiguated by their context or if they should be considered inconsistent.

6.1.2 NoSta-D

Analysis. For a preliminary analysis, we sampled the predictions for 4,519 types from NoSta-D. For 4,207 the predictions of the three models were identical, for 312 at least one model produced a unique lemma. As before, we will first discuss the unique errors, then the identical ones.

[3] Forms and lemmas can occur both in the training and evaluation data, as our types are tuples of *form*, *lemma*, *POS* and *morphological tags*.

Classes. During the annotation process, we noticed that the lemmatization style in NoSta-D is different from the one in TüBa-D/Z. Comparatives and superlatives, for instance, are reduced to their positive, whereas in TüBa-D/Z the correct lemma is the nominative singular of the respective degree. To account for cases where the produced lemma is correct according to the TüBa-D/Z guidelines (Telljohann et al., 2006), we also assign the correct class to these tokens, hence **Correct** does not solely reflect the share of predictions that matched the lemma but also those that we consider correct. It should be noted that this only eliminates false negatives but not false positives where a lemma matches the gold-standard which would be considered wrong in TüBa-D/Z style. For cases where the correct lemma was not certain to us we assign the **Undecided** class, this happened mostly to nominalized verbs that can only be disambiguated within context. An analysis of these errors remains for the future. To account for the prevalence of colloquial language, dialect forms and historic forms, we also introduced the class **Non-standard** in our analysis of NoSta-D.

	Correct	Undecided	Solvable	Ign. morph	Wr. morph	Separable	NE	Spelling	Non-standard	Unsolvable
Oh-Morph	38.14%	2.24%	17.30%	0.64%	1.60%	2.88%	0.32%	7.05%	21.47%	8.33%
Lemming-Base	37.82%	1.92%	15.06%	0.64%	6.09%	3.20%	0.32%	7.37%	21.15%	6.41%
Lemming-List	47.43%	2.88%	11.53%	0.32%	5.77%	2.88%	0.32%	4.17%	18.58%	6.08%

Table 10: Preliminary result of the analysis of 312 of the non-identical sampled predictions of the three models on NoSta-D. The biggest error sources are non-standard language and spelling, *Lemming-List* has the least of these errors. Abbreviations: *wr.*: wrong, *ign.*: ignored, *NE*: named entity.

Preliminary results. The preliminary results of the analysis of the unique predictions of NoSta-D are presented in Table 10. These results mostly confirm the model-specific findings of the analysis on TüBa-D/Z. Most likely due to their prevalence in the corpus, we find that errors connected to colloquial language, dialect forms, and non-standard spelling are the most common ones for all three models. Again, we find that *Lemming-List* produces the least errors of these classes. Moreover, we find an indication, stronger than on TüBa-D/Z, that *Oh-Morph* suffers less from erroneous morphological tags, possibly due to tagging errors being more frequent on the noisy data. There are more unsolvable cases than on TüBa-D/Z, most of these are related to the normalization of nicknames in the BeMaTaC sub-corpus of NoSta-D which we consider to be solvable only on a sentence or even extra-sentential level.

Correct	Undecided	Solvable	Ign. morph	Wr. morph	Separable	NE	Spelling	Non-standard	Unsolvable
92.42%	1.16%	0.33%	0.02%	0.09%	0.05%	0.5%	1.38%	2.41%	1.62%

Table 11: Preliminary result of the analysis of 4,207 of the identical sampled predictions on NoSta-D. Non-standard language and spelling are the biggest error sources. Abbreviations: *wr.*: wrong, *ign.*: ignored, *NE*: named entity.

Intersection. The analysis of the sampled unique predictions on NoSta-D is presented in Table 11. Since we also assigned the **Correct** class, we include it in the table. The first thing to notice is that unsolvable errors make up the second largest class. This is mostly explained by the aforementioned normalization of chat nicknames. Within the intersection we find that most errors are connected to non-standard variations like the historic forms within the Anselm sub-corpus. This confirms our finding on colloquial language in TüBa-D/Z. Further notable is again the big share of spelling related errors. We believe that the amount of errors related to the surface form of the words, e.g. non-standard spelling or spelling mistakes, overshadows the amount of other errors that could have been produced if the normalization step would not have failed already.

Out of vocabulary While we did not find a distinctive effect for out-of-vocabulary items as in TüBa-D/Z, we discover a known issue of Seq2Seq architectures, namely unseen input and output symbols. *Oh-Morph* is not equipped to deal with unknown characters and just dropped them in most cases. Lemming on the other hand, will only fail if the unknown character is part of an inflection, another advantage of its edit-trees. The issue for the Seq2Seq model might be tackled by introducing a copy item which replaces individual characters in parts where form and lemma align.

Inconsistencies. During our analysis of NoSta-D, we found several lemmas that were in fact inflected forms. A search for the lemma *Nägel* 'nails', for example, brings up 7 hits in the BeMaTaC sub-corpus. Further, we find that verbal adjectives are in some cases reduced to the verb they are derived from and in others to the nominative singular of the adjective. A more thorough examination remains for future work.

7 Conclusion

In this work, we have proposed a morphologically-informed variant of the Seq2Seq architecture for lemmatization. We evaluated its effectiveness on German and provided a detailed error analysis with an emphasis on robustness and show strengths and weaknesses of the respective models. The results show that the Seq2Seq architecture achieves competitive performance. More precisely we found that *Oh-Morph* is less prone to suffer from wrong morphological tags which might lead to a better ability to incorporate them into its predictions. *Lemming-List*, on the other hand, seems to benefit from its word list, as it indicates whether a candidate is well-formed or not. Lemming's big advantage here is that it is a classifier over a candidate set rather than a generative model. Generating the potential lemmas ahead of time allows to incorporate features of the lemma, such as spelling or it being present in a word list. A Seq2Seq system, in contrast, cannot recover from false predictions which might be a reason for its tendency to transfer spelling errors from form to lemma. Turning to out-of-vocabulary items, we find that *Lemming-List's* advantage on malformed forms leads to the worst performance on unknown lemmas, whereas *Oh-Morph* shows the best performance with both unknown forms and lemmas.

Since both spelling errors and unknown tokens are to be expected when processing noisy web-corpora, we believe that good performance on noisy input and unknown tokens should not be a contradiction. For future work we plan to tackle the issue with spelling errors, as we saw that almost 40% of the shared errors were due to these cases. Possible approaches include incorporating a word list or more global optimization algorithms like Minimum Risk Training (Shen et al., 2016) or MIXER (Ranzato et al., 2015). Work in this direction should explore the intersection of lemmatization and text normalization, possibly in a joint training scenario. Given the influence of wrong morphological tags we are also confident that improving on morphological tagging will yield better results. Here it could be worthwhile to explore whether jointly assigning morphological tags and lemmas yields the same improvements as Müller et al. (2015) and Chrupała (2006) report. Another possibility that should be explored, pointed out by an anonymous reviewer, is the investigation of the effect of frequency by training on tokens.

As we have found that *Lemming-List* and *Oh-Morph* have somewhat complementary strengths, other future work should look into a possible ensemble consisting of an edit-tree classifier and a Seq2Seq model. A first naive approach could add the Seq2Seq lemmas to the candidate set of Lemming. A natural follow-up would then explore neural classifiers for edit scripts, with

a potentially simpler architecture than that of a fully fledged Seq2Seq model. The idea is compelling as it would allow to include arbitrary features, including lemma features, while keeping the flexibility of a character based encoder, with, in contrast to the log-linear Lemming, feature interactions that come with neural networks.

Acknowledgments

Financial support for the research reported in this paper was provided by the German Research Foundation (DFG) as part of the Collaborative Research Center "The Construction of Meaning" (SFB 833), project A3. Moreover, we would like to thank Patricia Fischer for her extensive and helpful comments on an early version of this paper.

References

Baayen, R. H., Piepenbrock, R., and van H, R. (1993). The CELEX lexical data base on CD-ROM.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bergmanis, T. and Goldwater, S. (2018). Context sensitive neural lemmatization with lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1391–1400.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103.

Chrupała, G. (2006). Simple data-driven context-sensitive lemmatization. *Procesamiento del lenguaje natural, n^o 37 (sept. 2006), pp. 121-127*.

Chrupała, G., Dinu, G., and Van Genabith, J. (2008). Learning morphology with morfette.

Cotterell, R., Kirov, C., Sylak-Glassman, J., Walther, G., Vylomova, E., Xia, P., Faruqui, M., Kübler, S., Yarowsky, D., Eisner, J., et al. (2017). CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30.

Cotterell, R., Kirov, C., Sylak-Glassman, J., Yarowsky, D., Eisner, J., and Hulden, M. (2016). The SIGMORPHON 2016 shared task—morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22.

Dipper, S., Lüdeling, A., and Reznicek, M. (2013). NoSta-D: A corpus of German non-standard varieties. *Non-Standard Data Sources in Corpus-Based Research*, (5):69–76.

Dozat, T. and Manning, C. D. (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. Association for Computational Linguistics.

Durrett, G. and DeNero, J. (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.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. *CoRR*, abs/1706.02515.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Minnen, G., Caroll, J., and Pearce, D. (2001). Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Müller, T., Cotterell, R., Fraser, A., and Schütze, H. (2015). Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274.

Müller, T., Schmid, H., and Schütze, H. (2013). Efficient higher-order crfs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332.

Pütz, T. (2018). Neural Sequence to Sequence Lemmatization. B.A. thesis, Eberhard Karls Universität Tübingen. `https://uni-tuebingen.de/en/34984`.

Raffel, C., Luong, M.-T., Liu, P. J., Weiss, R. J., and Eck, D. (2017). Online and linear-time attention by enforcing monotonic alignments. In *International Conference on Machine Learning*, pages 2837–2846.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. (2015). Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Schmid, H., Fitschen, A., and Heid, U. (2004). SMOR: A German computational morphology covering derivation, composition and inflection. In *LREC*, pages 1–263. Lisbon.

Schnober, C., Eger, S., Dinh, E.-L. D., and Gurevych, I. (2016). Still not there? comparing traditional sequence-to-sequence models to encoder-decoder neural networks on monotone string translation tasks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1703–1714. The COLING 2016 Organizing Committee.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1073–1083.

Sennrich, R. and Haddow, B. (2016). Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, volume 1, pages 83–91.

Sennrich, R. and Kunz, B. (2014). Zmorge: A German morphological lexicon extracted from Wiktionary. In Chair), N. C. C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. (2016). Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1683–1692.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Telljohann, H., Hinrichs, E., Kübler, S., and Kübler, R. (2004). The TüBa-D/Z treebank: Annotating German with a context-free backbone. In *In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004*. Citeseer.

Telljohann, H., Hinrichs, E. W., Kübler, S., Zinsmeister, H., and Beck, K. (2006). Stylebook for the Tübingen treebank of written German (TüBa-D/Z). In *Seminar fur Sprachwissenschaft, Universitat Tubingen, Tubingen, Germany*.

Vinyals, O., Kaiser, Ł., Koo, T., Petrov, S., Sutskever, I., and Hinton, G. (2015). Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781.