Named-Entity Recognition for Norwegian

Bjarte Johansen
Digital Centre of Excellence
Equinor ASA
{bjajoh@equinor.com}

Abstract

NER is the task of recognizing and demarcating the segments of a document that are part of a name and which type of name it is. We use 4 different categories of names: Locations (LOC), miscellaneous (MISC), organizations (ORG), and persons (PER). Even though we employ state of the art methods—including sub-word embeddings—that work well for English, we are unable to reproduce the same success for the Norwegian written forms. However, our model performs better than any previous research on Norwegian text. The study also presents the first NER for Nynorsk. Lastly, we find that by combining Nynorsk and Bokmål into one training corpus we improve the performance of our model on both languages.

1 Introduction

NER is the task of recognizing and demarcating the segments of a document that are part of a name and which type of name it is. We use 4 different categories of names: Locations (LOC), miscellaneous (MISC), organizations (ORG), and persons (PER). Even though we employ state of the art methods—including sub-word embeddings—that work well for English, we are unable to reproduce the same success for the Norwegian written forms. However, our model performs better than any previous research on Norwegian text.

We also find that when we train on a combined corpus of Nynorsk and Bokmål, which we call Helnorsk, we get significantly better results (+5 percentage points) than if we train the models separately. We believe that this shows us, together with evidence provided by Velldal et al. (2017) that it is possible to use the similarities in the two written forms to produce better models than we would otherwise be able to when the models are trained separately. We discuss this further in section 7 and 8.

Previous research on NER for Norwegian has chosen a more granular approach to the categories of names and have included the categories "works" and "events". The reason we chose to exclude these two categories was firstly that international research on English and other languages mainly focus on the same categories as us—that means that it is easier for us to compare our research to what has been done for other languages.

Secondly, previous research on Norwegian NER does not implement the same type of model that we and international researchers have implemented. They focus solely on the task of recognizing what type of name an already segmented name is categorized as. Our research also includes the segmentation of the names as well. This makes it difficult to compare our research directly with theirs.

Using their tools would also prevent us from using the NER directly on new documents if we wanted to build new research on top of such a NER model. We would have to first segment the text through Named-Entity Chunking (NEC) and then run the their recognizer on the result from the NEC. Johansen (2015) does provide a chunker that performs well (>95% $F_{\beta=1}$ score) However, we want to see how well a model that use state-of-the-art algorithms developed for English will perform on Norwegian. These algorithms usually do chunking as an implicit step of the NER process.

In our study we show that our model performs better than all previous attempts at a Bokmål NER (> +5 percentage points). There are no other NER models for Nynorsk that we are aware of. We show that by combining Nynorsk and Bokmål, into what we call Helnorsk in our study, we get better results than if we train separate models for the two written forms. "Helnorsk" translates to "The whole of Norwegian", which is fitting as it combines both

Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa), pages 222–231
Turku, Finland, 30 September – 2 October, ©2019 Linköping University Electronic Press

of the official written forms.

The steps we take to present our study are to

1. Present related research on NER in section 2.
2. Introduce a new corpus which is tagged with named entities and their types in section 3.
3. Develop a sub-word embedding model for Nynorsk, Bokmål, and Helnorsk and implement a deep learning system designed to train a NER model based on a state-of-the-art English model in section 4.
4. Run experiments on Bokmål, Nynorsk, and Helnorsk to show how the model performs in section 5.
5. Discuss the results of the experiments in section 6.
6. Conclude on what we believe the experiments show us in section 7.
7. Present future research that we believe should be explored to answer some of the questions that we found at the end of this study in section 8.

2 Related research

Bick (2000) developed an early Danish NER base on constraint grammar parsing. They report an error rate of $\sim 5\%$. It is unclear how their measure relates to the more standard way of reporting accuracy with F-scores. Bick (2004) improved the first model and achieved an $F_{\beta=1}$ score of 93%. It is however unclear how they arrive at this score as they originally report on different error rates of the model and then say that these numbers translate to the given F score. They do not tell us how they translated these numbers.

Derczynski et al. (2014) worked on a NLP toolkit for danish based on the Stanford NER package that includes a NER part. They annotated the Copenhagen Dependency Treebank for person, location and organisation entities. However, they do not report on the performance of their tool.

Jónsdóttir (2003) did some early work on chunking and recognition for Norwegian Bokmål. They used a ruled-based approach through the use of constraint grammar rules. The approach did provide good recall scores (>90%) for NER, but the precision did not reach satisfactory results (<50%). Jónsdóttir does not provide the corresponding numbers for their NEC.

Nøklestad (2009) and Haaland (2008) also worked on named entities for Norwegian Bokmål texts. Nøklestad uses a Memory-Based Learning approach while Haaland uses Maximum Entropy Models. The main challenge with the approach implemented by Nøklestad and Haaland is that they only categorize names that are already chunked from the text. That means that they are dependent on a named-entity chunker to tell the categories of names in running text. Haaland provide a $F_{\beta=1}$ score of 81.36%, while Nøklestad achieve a score of 82.53%.

Husevåg (2016) explores the role of named entities in automatic indexing based on text in subtitles. They show that the distribution of named entities are not the same for all types of text and that Norwegian text has a significantly lower name density than English for non-fiction text. They also argue that NER is an important tool for indexing as named entities are a common search request.

Kokkinakis (2004) created a NER for Swedish and showed that they could get good results on a test corpus of 45962 tokens. They got a $F_{\beta=1}$ score of 90.50%.

Dalianis and Åström (2001) use a rule-based approach to NER for Swedish and show a $F_{\beta=1}$ score of 61%.

Mickelin (2013) also worked on NER for Swedish. They use SVM to train their model and achieve a $F_{\beta=1}$ score of 20%.

Olsson (2008) developed a tool for annotating NER data an showed that their tool decreases the number of documents an annotator needs to review and still get good results.

Kokkinakis et al. (2014) converted and adapted the NER described by Kokkinakis (2004) to the Helsinki Finite-State Transducer Technology platform (HFST). HFST is a pattern matching tool (Karttunen, 2011). Their NER tags 8 different categories: Person, location, organization, artifact, work, event, measure, and temporal. They report a precision of 79.02%, recall of 70.56%, and a $F_{\beta=1}$ score of 74.55%.

Kapočūtė-Dzikienė et al. (2013) use CRF to train a NER model for Lithuanian. They achieve an $F_{\beta=1}$ score of 89.5%.

Chiu and Nichols (2015) implemented NER for English using LSTM-BiRNNs, and is the research that we have tried to implement for Norwegian, except that we are using sub-word embeddings, represent the character and case information differently, and work with Norwegian text instead of English. We also combine two different written forms of the same language to increase performance.

Rama et al. (2018) present a new corpus consisting of Norwegian clinical trials annotated with entities and relationships. The entities are categorized into 10 different categories, while there are 5 different categories for relationships. They build two different models, one entity extraction model and one model for relationship extraction. The entity extraction model achieves a F_1 score of 84.1%. The relation extraction model achieves a F_1 score of 76.8%. They use SVMs for both models. The entities that they describe are not all fully *named* entities. They are also interested in finding family members addressed as, for example, "bestefar" (translation: grandfather) and nouns that refer to the patient in question, such as "pasienten" (translation: the patient).

Stadsnes (2018) trained and evaluated different word embeddings models for Norwegian and came to the conclusion that while fastText skipgram embeddings performed better when recognizing analogies, word2vec CBOW embeddings were better for synonym extraction. In section 5 we show that skipgrams work better for NER.

Peters et al. (2018) implemented NER for English using a novel approach they call ELMo, which "is a deep contextualized word representation that models both complex characteristics of word use (e.g. syntax and semantics) and how these uses vary across linguistic context (i.e. to model polysemy)." They achieve a $F_{\beta=1}$ score of 92.22% on English text.

3 Corpus

We introduce a newly tagged corpus with named entities for the task of NER of Norwegian text. It is a version of the Universal Dependency (UD) Treebank for both Bokmål and Nynorsk (UDN) where we tagged all proper nouns with their type according to our tagging scheme. UDN is a converted version of the Norwegian Dependency Treebank into the UD scheme (Øvrelid and Hohle, 2016).

Table 1 shows the distribution of the different types of text in the corpus. It consists of 82% newspaper texts, 7% government reports, 6% parliament transcripts, and 5% blogs (Solberg et al., 2014). Table 2 shows the number of names for each of the categories that the corpus has been tagged with. We chose to tag it with the same categories as the CONLL-2003 shared task for language-independent NER (Tjong Kim Sang and Buchholz, 2000): Location (LOC), miscellaneous [1] (MISC), organization (ORG), and person (PER). The corpus along with the source for the project can be found here: `https://github.com/ljos/navnkjenner`.

Resource	Percentage
Newspaper texts	82
Government reports	7
Parliament transcripts	6
Blogs	5

Table 1: Description of data set.

We chose this scheme despite previous research on NER for Norwegian has chosen a more granular approach (e.g. Haaland (2008); Jónsdóttir (2003); Nøklestad (2009)) This meant that we are to be able to more easily compare our NER tagger to taggers developed for English. Previous research studies on Norwegian text are also not solving the exact same problem as we are investigating for our research. They focus solely on categorizing named entities and do not also delineate them from the text at the same time. Having fewer categories also meant that an annotator could perform the tagging faster as there were fewer choices to make when they decided the category of a name.

There are however some constraints on our corpus. The corpus has only been tagged by one annotator in one pass. This means that there are probably mistakes which will affect the performance of the trained models. The type of deep learning model that is trained for this research can never be better than the input it receives. After some investigation of the UDN data set, we also decided to trust that all named entities were tagged in the original UDN corpus with the PROPN (proper noun) tag. It is entirely possible that some of the entities are tagged as nouns only, further degrading the performance.

During the tagging we noted that—especially for the Nynorsk part of the UDN corpus—not all parts of a name were always tagged as a proper noun. This is not necessarily wrong in a grammatical sense, but it does mean that the two written forms

[1] By "michellaneous" we mean a catch-all category where any named entity that does not belong in any of the other categories goes into this category.

Bokmål	Tokens	Sentences	LOC	MISC	ORG	PER	Total
Training	243894	15686	3241	498	3082	4113	10934
Development	36369	2410	409	113	476	617	1615
Test	29966	1939	420	90	317	564	1391
Total	310229	20035	4070	701	3875	5294	13940

Nynorsk	Tokens	Sentences	LOC	MISC	ORG	PER	Total
Training	245330	14174	3482	588	2601	3992	10663
Development	31250	1890	340	67	268	421	1096
Test	24773	1511	300	59	246	362	967
Total	301353	17575	4122	714	3115	4775	12726

Helnorsk	Tokens	Sentences	LOC	MISC	ORG	PER	Total
Training	489224	34170	6723	1086	5683	8105	21597
Development	67619	4300	749	180	744	1038	2711
Test	54739	3450	720	149	563	926	2358
Total	611582	41920	8192	1415	6990	10069	26666

Table 2: Number of names for each data set.

follow a slightly different grammatical UD tagging schema. Since the tagging of named entities was quite time consuming, we did not have time to investigate further or try to figure out how to correct any mistakes that were made in our named-entity tags or the PoS tags of the UDN corpus.

4 Method

For the NER tagger we chose to use the BIOES tagging scheme as other researchers report that the BIOES tagging scheme performs (marginally) better on this type of task (Lample et al., 2016). The BIOES tagging scheme uses 5 different tags, instead of the 3 of the IOB2 scheme. The tags are

B A token at the **beginning** of a sequence.

I A token **inside** a sequence.

O A token **outside** a sequence.

E A token at the **end** of a sequence.

S A **single** token representing a full sequence.

We tagged each of the tokens in our corpus with one of these tags and the corresponding class of that token. There is an example in table 3.

We then trained a CBOW and a skipgram embedding model for each of the language forms: Nynorsk, Bokmål, and Helnorsk. The models were trained on a cleaned and combined corpus consisting of texts from Wikipedia, the Norwegian News Corpus (Andersen and Hofland, 2012), and the Norwegian Dependency Treebank (Solberg et al., 2014). We used fastText to train the sub-word embeddings with a vectors size of 300 components with a minimum n-gram size of 2 and maximum of 5 for the sub-words (Bojanowski et al., 2017).

We created a gazetteer from the NER corpus by extracting all words that appear as part of a name in the corpus. The gazetteer is used as part of the input to the model so the model can tell if a token has been used as part of a given category of names in the past.

The model that we use is a bidirectional Recurrent Neural Network with a Long Short-Term Memory unit (biLSTM) and it is trained on sentences that we treat as sequences of words. Recurrent Neural Networks "are a family of neural networks for processing sequential data" (Goodfellow et al., 2016, Chap. 10).

For each word in the sequence, we create an input vector that consists of the sub-word embedding of the word, membership in the gazetteer, the sequence of the characters of the word, and the part-of-speech of the word.

A biLSTM is a recurrent neural network that walks the sequence in both the forward and backwards directions. Long Short-Term Memory units introduces "self-loops to produce paths where the gradient can flow for long durations" and thereby capturing long-term dependencies (Goodfellow et al., 2016, Chap. 10). Using biLSTMs allows

O	O	O	O	O	B-PER	I-PER	E-PER	O
Folk	er	så	opptatt	av	Karl	Ove	Knausgård	.
People	are	so	occupied	with	Karl	Ove	Knausgård	.

Table 3: Example of tagging a sequence that mentions a person.

us to capture information about each word from both the past and future words in the sentence.

We also train a character embedding as part of the model. The character embeddings for each character in a word is run through a 1-dimensional Convolutional Neural Network layer (CNN), and the output of the convolutional layer is pooled together by selecting the maximum value for each position in the vector from the character embeddings. By 1-dimensional we mean that the CNN operates on a view of the neighbouring characters in each word .

The convolutional layer is activated by a Rectified Linear Unit. It constrains the value its output to be 0 or greater and is used in many types of tasks from image classification to machine translation (Ramachandran et al., 2018).

CNNs are "neural networks that use convolution in place of general matrix multiplication" (Goodfellow et al., 2016) and are used in tasks such as image classification. Using a dense network for these types of tasks would require too many neurons to be possible to train in a reasonable amount of time. Instead of operating on every point of the image, each neuron in a CNN operates on a n-dimensional view of the input.

We use the sub-word embeddings, the part-of-speech, gazetteer information, and the pooled character embeddings as the input to the biLSTM layer.

The output of the biLSTM layer is then fed through a linearly-activated dense layer that reduces the dimensionality of the output from the biLSTM down to the number of tags in our vocabulary.

A dense layer is a neural network where every input to the layer is connected to every output of the layer (Mitchell, 1997). It still has a weight for every connection, an activation function, and a bias for every output in the network. Each node in the neural network calculates the affine transformation of the inputs where the inputs $\vec{x}$ are weighted by the kernel $\vec{w}$ and then summed together with a bias b. The bias makes it possible to improve the fit of the input of the activation function to the prdiction by altering the shape of the function. The bias is either set to a specific number like 1, or trained as one of the parameters of the network. The sum is then put through an activation function. The activation function acts as a decision boundary for the node.

The output of the dense layer is fed to a Linear-chain Conditional Random Field (CRF), that we use to calculate the log likelihoods of the predicted tags. We then use the CRF to calculate the most likely sequence given the evidence we have seen. The model can be found here: [redacted for review].

A CRF is used to classify sequences where the variables can be dependent on any other part of the sequence (Lafferty et al., 2001) like in a sentence. A CRF needs a takes a parameter vector that it uses for classification and is usually learned through an optimization algorithm, but in our case it is the output of the dense layer that we use as the parameter vector. In other words, the neural network decides the parameter vector for the CRF and then the CRF uses that to classify each token in the input.

Variable	Value
Batch size	100
Char. embed size	25
Conv. kernel	3
Pool size	53
Depth	1
Dropout	0.5
RNN hidden size	256
Learning rate	0.01

Table 4: Hyperparameter configuration of the model training.

We trained the model using the Adam optimizing algorithm on the cross entropy loss given the predicted likelihood for each tag. The cross entropy loss then provides how many bits are needed to represent the difference between the two distributions. Therefore, the smaller the difference, the more similar the distributions are.

We manually tested the training parameters, but because of time constraints we ended up using the hyperparameter configuration in table 4 as those were giving us the best results for the values that were tested.

Adam is an algorithm for "first-order gradient-based optimization of stochastic objective functions" (Kingma and Ba, 2014). It gets its name from the fact that it uses "adaptive moment estimation" to train the weights in the model based on the local moments, instead using the global moments as the estimated error.

The way the algorithm works is by calculating adaptive learning rates for different parameters by estimating the mean (the first moment) and the uncentred variance (the second moment).

In further detail, it first calculates the gradient for the stochastic objective of our loss function. Then it updates the first and second moment estimates based on the current timestep. It then uses the individual moment estimates of each gradient to calculate the updated parameters for the loss function. To update the network, it uses back-propagation of the errors through the network to update all the weights of the network.

To avoid the problem of exploding gradients in biLSTMs as described by Bengio et al. (1994), it is adviced to clip the gradients to the global norm, or to a max value, as suggested by Pascanu et al. (2013). The reason for this problem is that biLSTMs allow the network to keep information about the past for an unspecified amount of time. This results in "an explosion of the long term components, which can grow exponentially more than the short term ones" (Pascanu et al., 2013).

For each model we set a batch size of 100, a character embedding size of 25, the convolution kernel was 3, the max pooling of the convolution run was set to 53 wide and the biLSTM depth–or how many biLSTM layers there are—was 1. The dropout between layers was 50% and the hidden size of the RNN was 256 neurons. The learning rate for the ADAM optimizing algorithm was 0.01.

Dropout is a regularization technique that helps to reduce overfitting by holding out a percentage of the input to a neural network at random (Hinton et al., 2012). This forces each neuron in the network to detect a specific feature that can help the network give the correct prediction.

5 Results

The results from training the different models are displayed in table 5. We trained 4 different models. One for Bokmål, Nynorsk, and Helnorsk using the CBOW embedding model. It shows that the combined Helnorsk model performs better than either of the models trained on a single written form by $\sim$5 percentage points (p.p.) over both forms. We then trained a skipgram model for Helnorsk which performs $\sim$5 p.p. above the CBOW Helnorsk model.

In the end we end up with a $F_{\beta=1}$ score of 86.73%, with a precision of 87.22% and recall of 86.25% for the combined written form. The model performs slightly better on Bokmål with an $F_{\beta=1}$ score of 87.20%, precision of 87.93%, and recall of 86.48%. The same model has an $F_{\beta=1}$ score of 86.06% for Nynorsk, 86.20% precision, and 85.93% recall.

Written form	Precision	Recall	$F_{\beta=1}$
Bokmål, CBOW	80.03	73.47	76.61
Nynorsk, CBOW	77.86	68.04	72.62
Helnorsk, CBOW	84.42	76.33	80.17
H/Bokmål, CBOW	87.06	77.42	81.96
H/Nynorsk, CBOW	80.78	74.76	77.65
Helnorsk, SG	87.22	86.25	86.73
H/Bokmål, SG	87.93	86.48	87.20
H/Nynorsk, SG	86.20	85.93	86.06
Helnorsk, SG-g	86.69	85.96	86.32
H/Bokmål, SG-g	87.74	86.48	87.11
H/Nynorsk, SG-g	85.21	85.21	85.21

Table 5: Results of NER experiments. (CBOW = continous-bag-of-words, SG = skipgram, SG-g = skipgram with smaller gazetteer)

LOC	Nynorsk	Bokmål	Helnorsk
Precision	87.98	89.55	88.89
Recall	90.33	89.76	90.00
$F_{\beta=1}$	89.14	89.65	89.44
ORG			
Precision	81.63	80.06	80.74
Recall	81.30	82.33	81.88
$F_{\beta=1}$	81.46	81.18	81.31
MISC			
Precision	71.88	74.54	73.56
Recall	38.98	45.56	42.95
$F_{\beta=1}$	50.54	56.55	54.23
PER			
Precision	88.91	92.58	91.11
Recall	93.09	92.90	92.98
$F_{\beta=1}$	90.96	92.74	92.04

Table 6: Per name precision, recall, and F_1 score for the best performing Helnorsk model.

In table 6 the pr. name category results are displayed. There, it can be seen that it is especially the miscellaneous (MISC) category that through its recall score is driving the results down with a score of 42.95%. The precision is also low with a score of 73.56%.

The organisation (ORG) category also performs worse than the total score with an $F_{\beta=1}$ score of 81.31%. It is the location (LOC) category, with a $F_{\beta=1}$ score of 89.44%, and especially the person (PER) category with a $F_{\beta=1}$ score of 92.04%, that is pushing the over all score upwards.

During the writing of the paper we discovered a mistake in the experimental setup: We had included the names from the full corpus (the training *and* test data), instead of just the training data. This leaks information between the training and test steps and could in turn lead to overfitting the model to the test data. We were able to rerun the experiment without the names from the test data for the Helnorsk model. The results of that are reported in table 5 with the label "SG-g". With this model, the results reduce slightly over most of the measures (<1 p.p), except on the recall of Bokmål where it stays the same.

It is difficult to tell if the difference between the SG and SG-g experiments are because of some variation in the random initialization of the weights, random dropout between the layers, or some other variant. As we did not have time to control for these variables we still report the results of the model with the full gazetteer with the caveat that it includes data from the whole corpus. It could also be that because we use dropout, the gazetteer becomes an unreliable feature and is not used. In the future, we could test this through feature ablation testing—removing features from a model to see which features contribute the most to the performance of the model.

6 Discussion

When comparing the results from our research with that of other research that has been done on the Norwegian written forms, it is evident that our model performs significantly better than what has been shown before:

Haaland (2008) and Nøklestad (2009) shows a $F_{\beta=1}$ score of 81.36% and 82.53%, respectively, for Bokmål and we have a score of 87.20%; almost 5 p.p improvement over their results. However, the comparison is not completely fair. They only try to categorize already segmented names. Our research segments and categorizes the text as part of the same process.

Jónsdóttir (2003) shows a $F_{\beta=1}$ score of 60%. We cannot boast of the same precision that they have (90%) for Bokmål, but we are close with 87.93%. They do not provide any results for Nynorsk.

(Rama et al., 2018) developed an entity extraction model based on SVMs and got a $F_{\beta=1}$ score of 84.1% on a corpus of clinical texts. They are interested in finding nouns, and not only named entities, such as "bestefaren" (translation: the grandfather), and it is therefore difficult to compare our study with theirs.

Chiu and Nichols (2015) achieves a $F_{\beta=1}$ score of 91.62% on the CoNLL-2003 data set and 86.28% on the OntoNotes data set. Both are English data sets. The CoNLL-2003 data set is somewhat comparable to our data set on the number of entities and tokens. Their corpus has 35089 entities over 302811 tokens (Tjong Kim Sang and De Meulder, 2003), while ours has 26666 entities over 611582 tokens for the Helnorsk data set. The OntoNotes data set is 104151 over 1388955 tokens and is much larger than the data set we have available for Norwegian. We see here that the ratio between tokens and entities in OntoNotes is $\sim$7%, and in CoNLL-2003 it is $\sim$12%, while for the Helnorsk data the ratio is $\sim$4%.

Though the CoNLL-2003 data set uses BIO (or IOB2) tags and we use BIOES, this is not a problem as we are not comparing how well the model is labelling each word, but how well the model finds and categorizes named entities.

This supports the conclusion by Husevåg (2016) that Norwegian has a much lower density of named entities compared to English. Since deep learning models require large amounts of data to generalize effectively over the data set, it is possible that this is a problem for training a model for NER on Norwegian text.

We saw in table 6 that the worst performing name category is the miscellaneous category. This is also the category with the fewest named entities, showing us that lower amounts of data gives us worse performance. If one looks at how many names there are for each category, in table 2, and compare to the performance on each category, it shows that the score is higher if there are more examples of names.

Peters et al. (2018) is the latest state-of-the-art

NER for English, as of writing, and achieves a $F_{\beta=1}$ score of 92.22% on the CoNLL-2003 data set. Though we are not able to reach the same score, we are only trailing by $\sim$5 p.p. Right now, there are many avenues to try out for research on Norwegian text to reduce that gap. In section 8 we discuss the ideas that we believe are the most promising and the most immediate.

7 Conclusion

The results of this research show that it is possible to train a deep learning model to learn how to find named entities in Norwegian text and reach close to ($\sim$5 p.p.) the results of state-of-the-art models for English text. Our model achieves a $F_{\beta=1}$ score of 86.73 on the combined Bokmål and Nynorsk corpus (called Helnorsk).

We also show that it is plausible that Norwegian is harder to train for NER because Norwegian has a lower density of named entities compared to English.

We also show that we can get better performing models for both the written forms, Bokmål and Nynorsk, if we use (sub)word embeddings and train on a combined data set instead of training a separate model for each written form of the language. We do not know if this way of combining Nynorsk and Bokmål into one training set will transfer to other natural language tasks.

We do see some challenges like a worse result for Nynorsk compared to Bokmål, which we cannot immediately explain. However, Velldal et al. (2017) has shown similar results as us when they trained a PoS tagger using a combined corpus instead of treating the two written forms as distinct languages.

8 Future work

There are many possible avenues for improving on this research in the future. The first thing we would like to try would be to do a hyperparameter search to see if there are other parameter settings that could improve the results further. We should also perform ablation testing of the input features to see which of the features are the most important to the network. This could give us information about where we should focus our work to improve the model further.

The comparison between the Helnorsk data set and the Nynorsk/Bokmål data could also be improved. As of this paper, it is difficult to say if the improved scores are caused by having a larger data set that is good enough or if the combined data set is truly better. A way we could do this is to run the training on a selection of the Nynorsk and Bokmål data that has the same size as those data sets.

Next, we should investigate if we can train and use the ELMo embeddings presented by Peters et al. (2018) for Norwegian. They report a relative increase of 21% on NER for English using their new embedding model.

More time should be spent on analyzing and cleaning the corpus. For now, only 1 annotator has gone through and annotated the data set with NER tags.

We would also like to investigate why the miscellaneous category is performing so much worse than the other categories. This could be because we have more mistakes there or that the category is too broad; and it is difficult for the model to find a good delineation between the names in the category and the rest of the corpus.

We would also like to further test the hypothesis that a model trained on both written forms performs better than if we train two separate models. Is it just because we have more training data, and despite introducing noise, it performs better; or is it the model that is able to generalize better over the wider data set? Does the performance increase hold for other natural language processing tasks? Is it just Nynorsk and Bokmål that exhibits this behavior, or can we include other similar languages like Swedish and Danish? How close do the languages have to be to show this type of performance increase?

References

Gisle Andersen and Knut Hofland. 2012. Building a large corpus based on newspapers from the web. *Exploring Newspaper Language: Using the web to create and investigate a large corpus of modern Norwegian*, 49:1.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Eckhard Bick. 2000. Named entity recognition for danish. *I: Årbog for Nordisk Sprogteknologisk Forskningsprogram*, 2004.

Eckhard Bick. 2004. A named entity recognizer for danish. In *Fourth International Conference on Language Resources and Evaluation*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Jason P. C. Chiu and Eric Nichols. 2015. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*.

Hercules Dalianis and Erik Åström. 2001. Swenam—a swedish named entity recognizer. *Technical Report, TRITANA-P0113, IPLab-189, KTH NADA*.

Leon Derczynski, Camilla Vilhelmsen Field, and Kenneth S Bøgh. 2014. Dkie: Open source information extraction for danish. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 61–64.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Åsne Haaland. 2008. *A Maximum Entropy Approach to Proper Name Classification for Norwegian*. Ph.D. thesis, University of Oslo.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Anne-Stine Ruud Husevåg. 2016. Named entities in indexing: A case study of tv subtitles and metadata records. In *Networked Knowledge Organization Systems Workshop*, pages 48–58.

Bjarte Johansen. 2015. Named-entity chunking for norwegian text using support vector machines. *NIK: Norsk Informatikkonferanse*.

Andra Björk Jónsdóttir. 2003. *ARNER, what kind of name is that? - An automatic Rule-based Named Entity Recognizer for Norwegian*. Ph.D. thesis, University of Oslo.

Jurgita Kapočūtė-Dzikienė, Anders Nøklestad, Janne Bondi Johannessen, and Algis Krupavičius. 2013. Exploring features for named entity recognition in lithuanian text corpus. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013); May 22-24; 2013; Oslo University; Norway. NEALT Proceedings Series 16*, pages 73–88. Linköping University Electronic Press.

Lauri Karttunen. 2011. Beyond morphology: Pattern matching with fst. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 1–13. Springer.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Dimitrios Kokkinakis. 2004. Reducing the effect of name explosion. In *Proceedings of the LREC Workshop: Beyond Named Entity Recognition, Semantic Labelling for NLP tasks*, pages 1–6.

Dimitrios Kokkinakis, Jyrki Niemi, Sam Hardwick, Krister Lindén, and Lars Borin. 2014. Hfst-swener – a new ner resource for swedish. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann Publishers Inc.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Joel Mickelin. 2013. Named entity recognition with support vector machines. Master's thesis, KTH Royal Institute of Technology.

Tom M. Mitchell. 1997. *Machine Learning*. McGraw-Hill.

Anders Nøklestad. 2009. *A machine learning approach to anaphora resolution including named entity recognition, pp attachment disambiguation, and animacy detection*. Ph.D. thesis, University of Oslo.

Fredrik Olsson. 2008. *Bootstrapping named entity annotation by means of active machine learning: a method for creating corpora*. Ph.D. thesis, University of Gothenburg.

Lilja Øvrelid and Petter Hohle. 2016. Universal dependencies for norwegian. *Proceedings of the Ninth International Conference on Language Resources and Evaluation*.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Taraka Rama, Pål Brekke, Øystein Nytrø, and Lilja Øvrelid. 2018. Iterative development of family history annotation guidelines using a synthetic corpus of clinical text. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 111–121.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. 2018. Searching for activation functions.

Per Erik Solberg, Arne Skjærholt, Lilja Øvrelid, Kristin Hagen, and Janne Bondi Johannessen. 2014. The norwegian dependency treebank. In *Ninth International Conference on Language Resources and Evaluation*.

Cathrine Stadsnes. 2018. Evaluating semantic vectors for norwegian. Master's thesis, University of Oslo.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task: Chunking. In *Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning - Volume 7*, ConLL '00, pages 127–132. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Erik Velldal, Lilja Øvrelid, and Petter Hohle. 2017. Joint ud parsing of norwegian bokmål and nynorsk. In *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*, pages 1–10. Linköping University Electronic Press.