Part-of-Speech Tagging for Icelandic using a Bidirectional Long Short-Term Memory Model with combined Word and Character embeddings

Steinþór Steingrímsson
Reykjavik University
steinthor18@ru.is

Örvar Kárason
Reykjavik University
orvark13@ru.is

Abstract

Using bidirectional long short-term memory models (bi-LSTMs) for various natural language processing tasks, including part of speech tagging, has become popular in recent years. Much of this previous work has delt with small tagsets or morphologically simple languages. We evaluate using bi-LSTM models to tag Icelandic, a morphologically rich language using a relatively large tagset. We then compare the results of using a bi-LSTM model with combined word and character embeddings to traditional PoS taggers for Icelandic. We achieve better results than any other taggers not taking advantage of a morphological database, suggesting that by incorporating such data we are likely to outperform current state-of-the-art results for Icelandic.

1 Introduction

Part-of-Speech (PoS) tagging is an important step for many NLP applications, such as named entity recognition, word-sense disambiguation, sentiment analysis and question answering. Since PoS tagging is a sequential labeling task, it is reasonable to assume that recurrent neural networks (RNN), a variant of Deep Neural Networks (DNN), could be effective for the task. Bidirectional long short-term memory models (bi-LSTM), have in recent years proven good for various natural language processing (NLP) tasks including PoS tagging, see, for instance Ling et al. (2015); Wang et al. (2015). LSTMs are a variant of recurrent neural networks (RNN) where the cells are designed to prevent vanishing gradients (Hochreiter and Schmidhuber, 1997). Bidirectional LSTMs are an extension on general LSTMs that perform better on sequences where the complete input sequence is available. Two LSTMs are trained on the input sequence, the first as it is and the other on it reversed (Graves and Schmidhuber, 2005). Using sub-token representations or character embeddings was first used for tagging by Dos Santos and Zadrozny (2014). This entails not only examining the sequences of words in a sentence during training but also the sequences of characters within those words.

We consider the use of bi-LSTMs with both word and character embeddings for PoS tagging of a morphologically rich language, i.e. Icelandic, with a large tagset. Only a small portion of previous work using neural networks for PoS tagging has focused on languages with complex morphology and large tagsets.

Contributions: We evaluate employing both word embeddings and character embeddings, dynamic embedding selection, and a combination of both in bi-LSTMs on a morphologically rich language with a large tagset.

2 Tagging with bi-LSTMs

Recently PoS-tagging with bidirectional LSTMs has been achieving state-of-the-art performance (Ling et al., 2015; Wang et al., 2015). For morphologically complex languages, Plank et al. (2016) have shown that bi-LSTM models can work especially well, although they only show that on datasets with very coarse-grained tagsets (17 tags or less). Horsmann and Zesch (2017) replicate that study using a collection of corpora annotated with fine-grained tagsets of varying sizes. They confirm the results in general, but assert that to reach the state-of-the-art for the very large tagsets of morphologically rich languages, hand-crafted morphological lexicons are still necessary.

2.1 Implementation

We implemented a couple of LSTM PoS tagger variations. Our models were built using DyNet[1] which is a neural network library that uses dynamic computation graphs and is thus well suited to NLP tasks (Neubig et al., 2017). The computation graph is composed of expressions, which relate to the inputs and outputs of the network, as well as the parameters of the network. Using dynamic computation graph declarations entails creating graphs for each expression instead of creating one static graph initially and then having to pad or truncate data that does not match its size. The source code for our implementations is available from `https://github.com/orvark13/DEEPtagger`

2.2 Word and character embeddings

Word embeddings are multidimensional vector representations of words based on their context in training data, i.e. surrounding words. The vectors encode various aspects of word similarity and can therefore reflect their syntactic and semantic details in a somewhat indirect way.

Using character embeddings has been shown to significantly improve performance for handling of unknown words (e.g. Plank et al. 2016; Dos Santos and Zadrozny 2014). For each word both forward and backward expressions are generated, containing the sequence of characters in the word as well as word initial and word final markers. This opens up the possibility for the network to learn various morphological details.

For unknown words as well as unknown characters we used a vector initialized with zeros. In effect placing the unknown word or character in the middle of the multidimensional vector space. Using zero vectors resulted in improvements over all variations of randomly initialized vectors we tried out.

2.3 Dynamic embedding selection

The dynamic embedding selection model used word embeddings for words that have a higher frequency in the training than a given parameter value and character embeddings for the ones less frequent. The character embeddings are thus not only helping to improve the performance for unknown word but also very infrequent words where the word embedding vectors are likely not to have become representative of the word in question.

[1]The Dynamic Neural Network Toolkit, see `http://dynet.io`.

2.4 Combining word and character embeddings

The last model entailed concatenating word and character embeddings into a single vector. This provides the network with all the available data and lets it figure out the significant bits at each step. As the task is more complex this model takes longer to train than the dynamic embedding selection model.

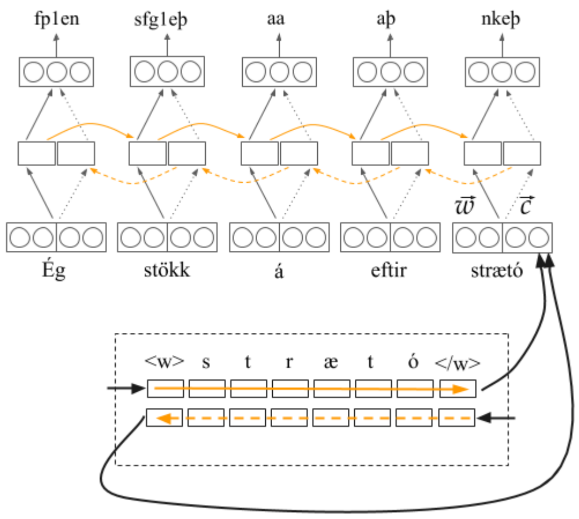

Figure 1: Combining word and character embeddings. Adapted from Plank et al. (2016).

Figure 1 shows a combined embeddings model as described in 2.4. Both word and character embeddings are used as input. The character embeddings for a given word are input into two LSTMs. The first LSTM reads the characters in the correct order, from left to right, and the other one in reverse order. The output from these LSTMs are concatenated to the word embedding and the concatenated vector is used as an input into another bidirectional LSTM. These bi-LSTMs return two vectors. They are concatenated and input into a hidden layer (omitted in the figure). The hidden layer feeds the output layer, which guesses the correct output, a PoS tag.

For our other models, either word embeddings or character embeddings would be input into our model, not the concatenation of both. That reduces training time significantly, but at the cost of accuracy as reported in Table 1.

	Acc.	Sent.	Known	Unkwn.	Epoch
Word emb.	87.78%	32.44%	93.42%	11.04%	40/40
Character emb.	86.38%	28.28%	88.16%	62.16%	26/40
Dynamic emb. selection	91.30%	41.54%	93.04%	**67.61%**	27/40
Combined emb.	**92.98%**	**48.01%**	**94.96%**	66.08%	286/300
Combined + pre-trained word emb.	92.71%	46.86%	94.76%	64.84%	40/40

Table 1: Accuracy results for the five models.

3 Experiments

We developed, trained and evaluated the following five variants of bi-LSMT models:

- Word embeddings only
- Character embeddings only
- Dynamic embedding selection
- Combined word and character embeddings
- Combined word and character embeddings with pre-trained word embeddings from an unannotated corpus

We experimented with a number of optimizers; RMSprop, Adaptive Movement Estimation (Adam) and three variants of Stochastic Gradient Descent (SGD). We found that a basic SGD optimizer achieved the best initial results for our task, given the other hyperparameters we experimented with. Thus we selected it for further training and testing. We also tried to use an SGD optimizer with momentum, but using it did not pan out. On the other hand an SGD optimizer that uses cyclical learning rates for training (Smith, 2015) seemed promising. It uses variable learning rate selected from a defined range. That entails defining a range of possible learning rate values is defined instead of a single fixed value. It would be interesting to experiment further with that optimizer.

Using pre-trained embeddings is a way to perform semi-supervised learning (or transfer learning). We tried initializing the word embeddings vectors with pre-trained embeddings, following Plank et al. (2016) which report significant benefits from that method. We used pre-trained Word2Vec embeddings (Mikolov et al., 2013) based on Icelandic texts containing approximately 500 million words, but trained for a different project. This seemed to give us a small boost in the first few epochs but no other gains and thus we decided on not using them. Pre-trained embeddings are discussed further in section 4.

3.1 Dataset

PoS taggers developed for Icelandic so far have all been trained and tested on The Icelandic Frequency Dictionary (IFD) corpus (Pind et al., 1991) containing about 590.000 tokens. As with the other taggers referenced in this report we use the so-called *corrected version* of the corpus, with a reduced tagset (565 tags) and ten-fold split from (Loftsson et al., 2009).[2]

3.2 Results

All models used the following hyperparameters, unless otherwise specified: SGD trainer, softmax negative log likelihood loss, learning rate of 0.11, 128 dimensions for hidden states and 128 for embeddings with Gaussian noise of 0.1. Following Adams (1979) we used 42 as the seed to both Python's and DyNet's pseudo-random number generators. Results from the different models tested are shown in Table 1. The hyperparameters for all models were based on the settings that gave the best results for the combined word and character embeddings model. Somewhat higher accuracy scores could very likely be achieved for the other models by tuning the hyperparameters individually for them.

Figure 2 shows the accuracy of all of the models we trained, except for the combined model initialized with pre-trained word embeddings. We leave that one out as it is almost exactly the same as the line for the combined model without the embeddings, except for a few of the first epochs. The picture shows that the model for character embeddings, as well as the dynamic embedding selection model have started overfitting. Our best model, the combined word and characters model, is still improving and gaining accuracy at the end of the initial 40 epoch runs. We therefore decided to continue training it and see how good it would get before it started to overfit. We trained the model for

[2]The IFD corpus was downloaded from `http://malfong.is/?pg=ordtidnibok`.

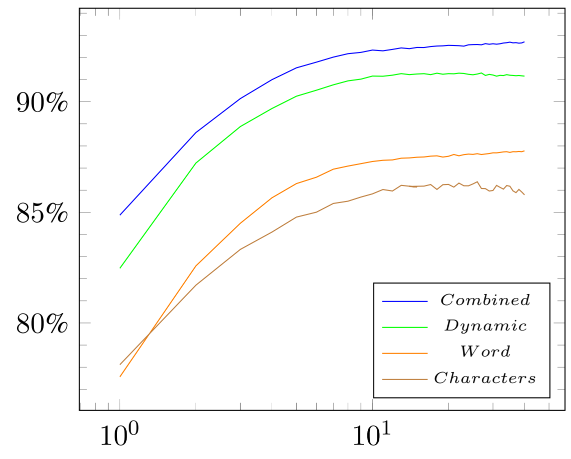

Figure 2: Accuracy (y-axis) of four models at each epoch (x-axis, log. scale) up to epoch 40.

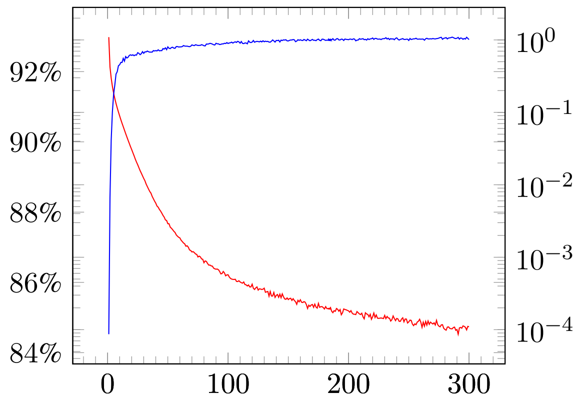

Figure 3: Accuracy (left y-axis) and average loss (right y-axis) at each epoch for the combined embeddings model.

300 epochs. Training and executing the 10-fold validation for 300 epochs took 46 hours on a 3.2 GHz i7-8700 processor with 6 cores and 12 logical cores. Each fold was allocated 1GB of memory.

In epoch 286 our model reached its best results, 92.98% overall word accuracy. As is evident from figure 3, the model had not start overfitting by epoch 300, so it might achieve even better accuracy if trained longer. These are state-of-the-art results and surpass all other taggers trained on this dataset except for those that take advantage of the declension paradigms from DMII.[3] Furthermore our model outperforms all other taggers on correctly tagging known words, i.e. words seen during training, whether they use the DMII or not.

A comparison of our highest scoring model to other generic taggers is shown in Table 2. All results are fully comparable as they are based upon the exactly the same cross validation split of IFD.

4 Related work

Various taggers have been developed for Icelandic, including a rule-based tagger and a hybrid tagger. Many, including the ones that have achieved the best performance, have been data-driven (Loftsson and Östling, 2013). The best results reported for these taggers, tested on the same dataset used in our experiments, were discussed in section 3. The work reported here is the first attempt at creating a DNN tagger for Icelandic.

In the last few years a number of papers have been written on PoS tagging using DNNs, particularly LSTMs (Ling et al., 2015; Wang et al., 2015; Plank et al., 2016). Recently work has also been reported on using CNNs, giving good results for

[3]It had already surpassed general taggers setups that were not employing any extraneous morphology data by epoch 65.

	Acc.	Known	Unkwn.
TnT	90.36%	91.74%	71.60%
Stagger	91.29%	93.11%	62.29%
Stagger + LF	91.42%	93.26%	66.06%
Nidir+WC+CT	92.06%	93.70%	69.74%
HMM+IceTagger+HMM	92.73%	93.84%	**77.47%**
Stagger + LF + IceMorphy	92.82%	93.97%	77.03%
Combined emb. model	92.98%	**94.96%**	66.08%
HMM+IceTagger+HMM + DMII	93.48%	93.85%	60.50%
Stagger + LF + IceMorphy + DMII	93.70%	94.02%	61.45%
Stagger + LF + IceMorphy + DMII + WE	**93.84%**	94.15%	61.99%

Table 2: Comparison to other taggers for Icelandic.

the Universal Dependencies tagset (17 tags) (Yu et al., 2017). Most of these taggers use combined character embeddings and word embeddings, as our best performing model did.

It is common to take advantage of pre-trained word embeddings trained on large corpora, external to the training and testing data. Such embeddings are often used for initialization, as described in Plank et al. (2016). As these embeddings are usually created by algorithms that are very different to bi-LSTMs, their applicability and usefulness may be limited. But other methods of taking advantage of pre-trained word embeddings have been proposed. Wang et al. (2015) introduced a novel method for pre-training word embeddings for use with bi-LSTMs and gained considerable accuracy on a PoS tagging task.

5 Conclusions and further work

We have shown that bi-LSTM models with combined word and character embeddings can achieve state-of-the-art performance in PoS tagging of Icelandic texts, when compared to taggers not taking advantage of the morphological database, DMII. We experimented with different optimizers, learning rate, and noise. Although our experiments gave us rough ideas of the range we should be looking at, experimenting further with other optimizers and fine-tuning the hyperparameters is very likely to result in improved accuracy. Also, we did not use dropout, but that has been reported to improve performance in some bi-LSTM PoS taggers, see e.g. Vaswani et al. (2016) and Cimino and Dell'Orletta (2016).

Pre-trained embeddings did not add any benefits to our model, but they have been shown to be beneficial, especially if they are trained for use with bi-LSTMs. It could be worthwhile to pre-train word embeddings, using the Icelandic Gigaword Corpora (IGC) of 1.2 billion unannotated words (Steingrímsson et al., 2018), and use that as described in Wang et al. (2015). To compare to our model and to explore whether other types of DNNs can be competitive to a bi-LSTM model, creating a PoS tagger using only CNNs for both word and character embeddings could lead to interesting results.

Tagging accuracy could most likely be improved by finding a way to include the declension paradigms from DMII in the model. That data is the only real advantage that other taggers have as is evident from the tagger comparison in table 2. The DMII could at least be combined with the IGC corpus to augment the training data.

To try to understand better the strengths and weaknesses of bi-LSTM models in relation to other taggers an analysis of the errors it makes needs to be done. This entails examining what types of errors it makes and why for both the known and the unknown words category. It would be interesting to compare them to the kind of errors other taggers make, for instance the results of the error analysis in Loftsson and Östling (2013). That could furthermore help in determining how the model's performance could be improved in the future.

References

Adams, D. (1979). *The Hitchhiker's Guide to the Galaxy*. Pan Books, London, UK.

Cimino, A. and Dell'Orletta, F. (2016). Building the state-of-the-art in POS tagging of italian tweets. In *CLiC-it/EVALITA*, volume 1749 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Dos Santos, C. N. and Zadrozny, B. (2014). Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1818–II–1826. JMLR.org.

Graves, A. and Schmidhuber, J. (2005). 2005 special issue: Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Netw.*, 18(5-6):602–610.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Horsmann, T. and Zesch, T. (2017). Do LSTMs really work so well for PoS tagging? - A replication study. In *EMNLP*, pages 727–736. Association for Computational Linguistics.

Ling, W., Luís, T., Marujo, L., Astudillo, R. F., Amir, S., Dyer, C., Black, A. W., and Trancoso, I. (2015). Finding function in form: Compositional character models for open vocabulary word representation. *CoRR*, abs/1508.02096.

Loftsson, H., Kramarczyk, I., Helgadóttir, S., and Rögnvaldsson, E. (2009). Improving the pos tagging accuracy of icelandic text. In *NODALIDA*, pages 103–110. Northern European Association for Language Technology (NEALT).

Loftsson, H. and Östling, R. (2013). Tagging a morphologically complex language using an averaged perceptron tagger: The case of Icelandic. In *Proceedings of the 19th Nordic Conference on Computational Linguistics (NODALIDA 2013)*, NEALT Proceedings Series, pages 105–119, Oslo, Norway.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Neubig, G., Dyer, C., Goldberg, Y., Matthews, A., Ammar, W., Anastasopoulos, A., Ballesteros, M., Chiang, D., Clothiaux, D., Cohn, T., Duh, K., Faruqui, M., Gan, C., Garrette, D., Ji, Y., Kong, L., Kuncoro, A., Kumar, G., Malaviya, C., Michel, P., Oda, Y., Richardson, M., Saphra, N., Swayamdipta, S., and Yin, P. (2017). Dynet: The dynamic neural network toolkit. *CoRR*, abs/1701.03980.

Pind, J., Magnússon, F., and Briem, S. (1991). *Íslensk orðtíðnibók*. Orðabók Háskólans.

Plank, B., Søgaard, A., and Goldberg, Y. (2016). Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *CoRR*, abs/1604.05529.

Smith, L. N. (2015). No more pesky learning rate guessing games. *CoRR*, abs/1506.01186.

Steingrímsson, S., Helgadóttir, S., Rögnvaldsson, E., Barkarson, S., and Gudnason, J. (2018). Risamálheild: A Very Large Icelandic Text Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Vaswani, A., Bisk, Y., Sagae, K., and Musa, R. (2016). Supertagging with lstms. In *HLT-NAACL*, pages 232–237. The Association for Computational Linguistics.

Wang, P., Qian, Y., Soong, F. K., He, L., and Zhao, H. (2015). Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. *CoRR*, abs/1510.06168.

Yu, X., Falenska, A., and Vu, N. T. (2017). A general-purpose tagger with convolutional neural networks. *CoRR*, abs/1706.01723.