Human Language Technologies – The Baltic Perspective
I. Skadiņa and R. Rozis (Eds.)

doi:10.3233/978-1-61499-701-6-160

Deep Neural Learning Approaches for Latvian Morphological Tagging

Pēteris PAIKENS [1]
University of Latvia, Institute of Mathematics and Computer Science

Abstract. This paper describes ongoing research on improvements of morphological analysis, disambiguation and POS tagging for the Latvian language. Authors apply recent advances in sequential tagging with neural networks and word embeddings calculated from unlabeled corpus to improve morphological tagging accuracy. These approaches allow to reduce the fine-grained morphological tag word error rate from 7.9% of earlier best systems to 6.2%, and coarse-grained POS tag error rate from 3.6% to 2.2%.

Keywords. morphology, tagging, deep learning, neural networks

1. Introduction

Morphological analysis and tagging is a commonly required key stage in most natural language processing systems, especially for morphologically rich languages such as Latvian. Currently various morphological taggers are available for Latvian, but their accuracy lags behind the larger languages such as English. While it's reasonable to expect lower accuracy to distinguish between the many tags possible in a morphologically rich language, even for the coarse part of speech categories the best previously reported accuracy scores for Latvian have an error rate twice as large as the state of the art taggers for English – 5% vs 2.5% [1,2].

This is caused in part by the comparably much smaller amount of available annotated training data. However, recent advances in deep neural network machine learning have not only shown the potential to improve supervised learning tasks, but also can learn powerful representations from unlabeled data, e.g. word embeddings [3] highlighting one possibility to partly cross this accuracy gap.

In this paper we describe the ongoing experiments to apply neural network based approaches to the task of fine-grained morphological tagging of Latvian text. In addition to the linguistic resources used in earlier systems – annotated corpora, lexical resources and output of a rule-based morphological analyzer – we now also augment the system with additional word embedding data from a large unlabeled corpus [4]. In order to evaluate these results, we compare the new system with the current state-of-art taggers publicly available for Latvian.

[1] Corresponding Author: Email: peteris@ailab.lv

2. Problem Description

For the purposes of this task, we attempt to solve the problem of fine-grained morphological tagging – obtaining a tag that specifies the morphosyntactic properties of each word, while also evaluating the accuracy of the coarse-grained POS tagging.

We implement the following hypothetical improvements in order to evaluate their effect on the accuracy of morphological analysis of Latvian:

- Word embedding data, calculated from a large untagged corpus;
- Various neural network approaches – convolutional neural networks, bidirectional LSTM networks with a CRF layer which has shown excellent results for English [2] and 'wide and deep' structures [5];
- Different representations of morphosyntactic information – including data from paradigm-based morphological analyzer and replacing the classic approach of distinct tags with separate sets of output neurons for each morphosyntactic property, trained together.

3. Related Work and Evaluation Methodology

Current published work on Latvian morphological tagging includes two comparable taggers. One of baseline systems is a conditional Markov model statistical tagger based on Stanford CoreNLP system [6] as described in [7], and the other is based on averaged perceptron as described in [1]. The source code of both these systems is available on GitHub with a permissive license, and their accuracy is comparable – [1] reports 93.60% vs 93.67% accuracy scores on the same set of test data.

There is also earlier work that has been used in Tilde proprietary systems [8], but that is closed source and has been superseded by the newer systems, so it was not replicated and evaluated in this paper.

Current most relevant related work for tagging methods, achieving best results when evaluated on standard English datasets, is the research on LSTM-CRF combination [2]. There is an interesting recent implementation [9] that claims even better results, but at the moment of writing this paper the full details are not yet available.

3.1. Training data

For training and evaluation, we use the current versions of data from the contemporary balanced corpus of Latvian [10] and the Latvian treebank [11]. The designated split of data contains 95 012 tokens as the training corpus and 7 293 tokens as development corpus for tuning and testing the system, and for the work-in-progress evaluations and comparisons of various strategies. A separate evaluation corpus of 7 020 tokens was set aside and used at article submission time for the final evaluation and system comparison.

The data is split in these partitions on a per-document basis, as there is significant intra-document overlap of rare vocabulary and proper nouns, which generally are harder to analyze, and in sentence-based randomized splitting those words are shared between training and evaluation data. Because of this effect, document-based split of training and evaluation data would be a more accurate metric of how the taggers would generalize to new documents. The sentence-based randomization produces an artificially elevated metric, because the system has seen the majority of every document during training.

Due to this change in training and testing data, the numeric results are rather different and not directly comparable with other papers using earlier versions of the same corpus, so the earlier methods were also re-trained and re-evaluated on the current test set.

3.2. Baseline systems

The new results are compared with the two existing systems described above – Paikens-2012 and Ņikiforovs-2015. We use the latest version of code as available on GitHub, but re-train the models on the abovementioned set of training data to ensure a fair comparison. This test set appears to more difficult in part due to the change from sentence-based split to document-based split between training and evaluation data and the numeric results are not directly comparable with earlier papers.

In addition, we also calculate a naïve baseline, obtaining by simply picking the most frequently seen tag out of the tag candidates supplied by the morphological analyzer.

4. System Architecture

For the purposes of this paper, a large variety of neural network structures were tested during system development, but limiting all of them to pure neural network architectures with no post-processing. All experiments shared a common structure of input and output (evaluation) data and were implemented in Tensorflow for GPU-based machine learning.

For input, we use the following features:

- a one-hot encoding of the word form according to the vocabulary of training corpus with rare words treated as out of vocabulary;
- pre-calculated word embedding model [4];
- one-hot encodings of suffix letter n-grams up to length of 4;
- an n-hot vector showing which of the possible candidates for morphosyntactic tags are considered valid for this word, taken from a morphological analyzer based on lexicon and inflectional paradigms [12].

For output, we considered three different vector encodings – a one-hot vector of the possible coarse-grained part of speech categories (13 options), a one-hot vector of the fine-grained morphosyntactic tags (430 options), and an encoding representing each possible morphological attribute-value pair separately (70 elements); functionally equivalent to the fine-grained tag as each can be constructed from the other.

The currently best performing system (labeled “Full NN system” in evaluation) is a combination of various elements with the structure illustrated in Figure 1. It starts with fully connected neural network layers calculating a compressed representation of the comparably wide (~5000 units each) word form and suffix encodings, followed by a drop-out layer to facilitate generalization. This is concatenated together with the other input vectors and fed to a convolution layer that combines features from neighboring words to capture the close-range relations. Convolution window size of just 3 words was found sufficient, as farther relations are captured by a bidirectional layer of long short-term memory (LSTM) cells as initially proposed by [13], thus encoding both the forward and backward context. The final classification is done by a logistic function on the output of LSTM layer (after dropout) combined with the full, wide content of all input features as suggested by [5]. A concatenation of all three output types is used in training to minimize the cross-entropy between network output and expected values using Adam

optimizer algorithm [14] and applying standard regularization to network coefficients. The network converges in 20 epochs in less than 2 hours on a NVidia TitanX GPU based system.

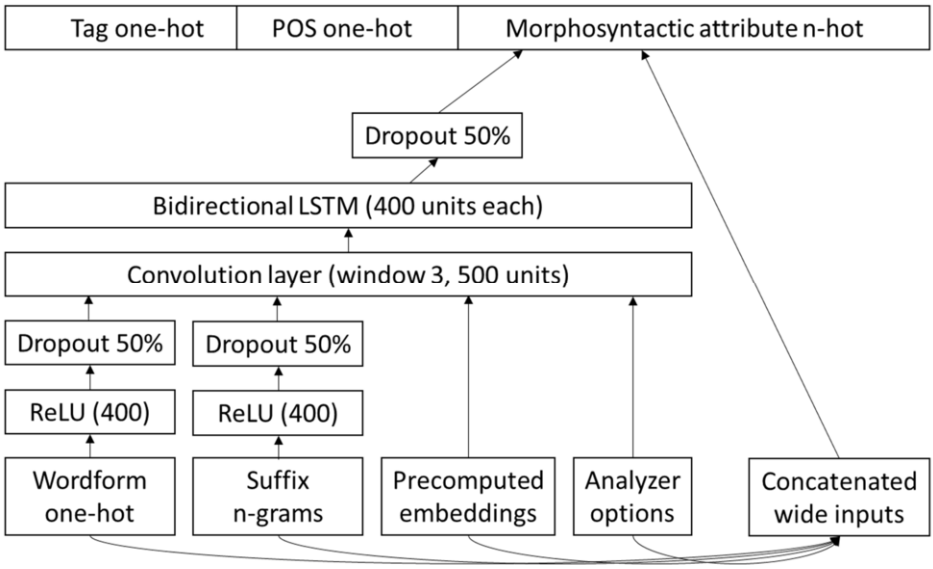

Figure 1. Network structure.

We also evaluate a minimalistic neural network structure, consisting of the abovementioned input layers, a single bidirectional LSTM layer with 200 cells, and the logistic output layer on top of that.

A large variety of other network structures were explored in experiments, but not exhaustively evaluated to verify and prove the effects of each separate factor. Nonetheless, the following observations and experience may be useful to the reader.

The choice of output encoding was highly significant. Using only the one-hot representation of tags (which seems to be the most commonly used approach in literature) without the separate attribute-value encoding lost about 1 full percentage point of accuracy.

Deeper network architectures beyond the proposed structure did not improve accuracy. We performed numerous experiments to explore various depths (up to 12) and layouts of recurrent and fully connected layers but these yielded the same or lower accuracy despite a much higher learning time or, in some configurations, performed significantly worse due to overfitting issues.

From the perspective of accuracy, the initial ReLU (rectified linear unit) layers after word form and n-gram encoding could better be replaced with a layer over the whole input vector set, however, they were necessary for performance reasons as the combination of wide inputs (11000-25000 neurons per word depending on vocabulary filtering) with larger sizes of further recurrent or convolutional layers result in operations that are impractical to train even on current top-end GPUs due to memory limitations.

5. Evaluation and Error Analysis

We compare the developed system against the baseline systems described in section 3.2, re-training them on the same set of updated corpora. The evaluation is shown in Table 1. In addition, we also consider three limited options:

- A much simpler NN model, consisting of only a single LSTM layer with 200 units between the input and output layers described earlier;
- A system which omits the morphological analyzer information while otherwise being identical to the full recommended system.
- A system trained without the attribute-value output, using only tag and POS.

Table 1. System evaluation.

System	Full tag accuracy	POS accuracy
Naïve baseline	71.9%	88.6%
Paikens-2012	91.4%	95.1%
Ņikiforovs-2015	92.1%	96.4%
Simple NN model	93.2%	97.6%
No analyzer	92.8%	97.7%
No attribute encoding	92.7%	97.7%
Full NN system	**93.8%**	**97.8%**

When run on a dataset with per-sentence split of training and evaluation data, the same dataset used in earlier experiments [1,7], the full NN system scores 95.4% for the full tag accuracy and 98.3% for POS accuracy. However, we don't consider those metrics as appropriate for evaluation because of issues described in section 3.1.

After performing the evaluation, a classification of errors of the best performing system on the test set was performed. The most popular errors (repeating 10 times or more) are shown on Table 2 and the per-feature error rates are shown in Table 3. Words that are out of vocabulary (with respect to training corpus) were found to have just slightly lower accuracy than average – 91.1% for full tag and 96.4% for POS.

Table 2. Popular errors.

Feature	Predicted value	Annotated value	Number of cases
Number	Singular	Plural	87
Number	Plural	Singular	42
Case	Genitive	Accusative	35
Case	Accusative	Genitive	33
Gender	Feminine	Masculine	32
Gender	Masculine	Feminine	32
Case	Genitive	Nominative	25
Case	Nominative	Genitive	20
POS	Residual	Noun	17
POS	Noun	Abbreviation	16
Definiteness	Definite	Indefinite	14
POS	Adjective	Verb	12
POS	Verb	Adjective	11
POS	Noun	Residual	10
POS	Residual	Abbreviation	10

As in earlier systems, the most popular error is the confusion between singular accusative and plural genitive, which are homoforms for many nouns and adjectives and whose disambiguation requires determining the case of a long noun phrase. The tagging errors for gender are in cases of contextual gender of pronouns and participles, where determining the 'correct' gender requires deciding to which noun this word refers.

The part of speech errors, on the other hand, seem to be caused by problems in corpus annotation. Names of foreign companies in newswire documents are variously tagged as inflexive nouns, residuals (foreign words) or abbreviations, causing confusion in such cases; and words which morphologically are derived from verbs but have obtained an independent adjective meaning are also inconsistently tagged as either verbs (participles) or adjectives, and thus show up as tagging errors.

Table 3. Feature error rates.

Feature	Error rate (for POS having this feature)
Part of speech	2.2%
Case	4.2%
Number	3.1%
Definiteness	3.0%
Pronoun type	2.3%
Gender	1.8%
Verb mood	0.6%
Residual type	0.4%

6. Conclusions and Future Work

Current experiments already noticeably outperform the baseline systems, showing that the approach is viable for improving morphosyntactic analysis of Latvian language, obtaining a significant increase in accuracy – the 1.7 percentage point improvement in tag accuracy amounts to a word error rate reduction from 7.9% to 6.2%, solving 20% of earlier system errors.

As in many use cases the next step in text processing is syntactic parsing, the dominant types of errors raise a peculiar Catch-22 situation – correct morphological tagging in these situation requires knowing the correct syntactic interpretation, while syntactic parsing requires morphological information and in these situations receiving wrong tags would result also in wrong syntactic dependencies. This suggests that further improvements in accuracy of morphological tagging might require doing syntactic parsing at the same time, as done by e.g. SyntaxNet [15].

As future work, it would be interesting to explore possibilities for replacing the morphological analyzer data with a character-level recurrent neural network, attempting to learn from unlabeled corpus the information that is currently taken from inflectional paradigms and lexical resources. Currently the system is usable without the morphological analyzer data, but suffers a noticeable decrease in tag accuracy.

It is interesting to note the surprisingly large effect of output representation as separate attributes instead of a list of tags. It may be worth exploring this effect in a more focused manner and check if it also holds for other morphologically rich languages.

The code and data for the final experimental system is freely available in GitHub at https://github.com/PeterisP/tf-morphotagger. Additional future work is expected in packaging this tagger for public use. While the earlier systems were easily distributable as Java and C# packages respectively, distributing Tensorflow systems to nontechnical end-users in a convenient way is difficult.

Acknowledgements

This research has been supported by Latvian State Research Programme SOPHIS (Project No. 2).

References

[1] Ņikiforovs, P. *Latviešu valodas morfosintaktiskais marķētājs.* Bachelor thesis, University of Latvia, 2015.
[2] Huang, Z., Xu, W., Yu, L. *Bidirectional LSTM-CRF Models for Sequence Tagging.* arXiv preprint arXiv:1508.01991 [cs.CL], 2015.
[3] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS)*, 2013.
[4] Znotiņš A. Word Embeddings for Latvian Natural Language Processing Tools. In this volume, 2016.
[5] Cheng, H. T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., ... & Anil, R. Wide & Deep Learning for Recommender Systems. arXiv preprint arXiv:1606.07792, 2016.
[6] Toutanova K., Klein D., Manning C.D. and Singer Y. Feature-Rich Part-of- Speech Tagging with a Cyclic Dependency Network. *Proceedings of HLT-NAACL* (2003), 252–259.
[7] Paikens, P., Rituma, L., and Pretkalniņa, L. Morphological analysis with limited resources: Latvian example. *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA)* (2013), 267– 277.
[8] Pinnis, M. and Goba, K. Maximum Entropy Model for Disambiguation of Rich Morphological Tags. *Systems and Frameworks for Computational Morphology, Communications in Computer and Information Science, 1, Volume 100, The 2nd Workshop on Systems and Frameworks for Computational Morphology (SFCM2011)*, Heidelberg, Springer (2011) 14–22.
[9] Choi, J.D. Dynamic Feature Induction: The Last Gist to the State-of-the-Art. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (NAACL'16)* 2016.
[10] Levāne-Petrova K. Morfoloģiski marķēta valodas korpusa izmantošana valodas izpētē. *"Vārds un tā pētīšanas aspekti": Rakstu krājums 15(1)*, Liepāja, LiePA (2011) 187–193.
[11] Pretkalniņa L., Nešpore G., Levāne-Petrova K., and Saulīte B. Towards a Latvian Treebank. *Actas del 3 Congreso Internacional de Lingüística de Corpus. Tecnologias de la Información y las Comunicaciones: Presente y Futuro en el Análisis de Corpus*, eds. Candel Mora M.Á., Carrió Pastor M., (2011) 119–127
[12] Paikens, P. Lexicon-based morphological analysis of Latvian language. *Proceedings of 3rd Baltic Conference on Human Language Technologies (HLT 2007)*, (2007) 235–240.
[13] Graves, A., Mohamed, A., Hinton, G. Speech Recognition with Deep Recurrent Neural Networks. arXiv preprint arXiv:1303.5778 [cs.NE], 2013.
[14] Kingma, D., & Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[15] Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., ... & Collins, M. Globally normalized transition-based neural networks. arXiv preprint arXiv:1603.06042, 2016.