Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200618

Lexicon-Enhanced Neural Lemmatization for Estonian

Kirill MILINTSEVICH [1] and Kairit SIRTS
Institute of Computer Science, University of Tartu, Tartu, Estonia

Abstract. We propose a novel approach for Estonian lemmatization that enriches the seq2seq neural lemmatization model with lemma candidates generated by the rule-based VABAMORF morphological analyser. In this way, the neural decoder can benefit from the additional input considering that it has a high likelihood of including the correct lemma. We develop our model by stacking two interconnected layers of attention in the decoder—one attending to the input word and another to the candidates obtained from the morphological analyser. We show that the lexicon-enhanced model achieves statistically significant improvements in lemmatization compared to baseline models not utilizing additional lemma information and achieves a new best result on lemmatization on the Estonian UD test set.

Keywords. Lemmatization, seq2seq model, Estonian, morphological analyser

1. Introduction

High quality lemmatization can drastically improve the quality of other more high-level NLP tasks such as information extraction [1] or named entity recognition [2]. This is even more important for languages with a rich word inflection paradigm like Estonian. Mapping word forms to their lemma greatly reduces the size of the training vocabulary and improves the learning capabilities of the models trained on such data.

State-of-the-art lemmatization systems are nowadays based on sequence-to-sequence neural architectures that can capture contextual word similarities better than purely statistical models. They are also not dependent on fixed lexicons and rule-based systems. Neural network based lemmatization systems have already achieved very high results. For instance, the best Estonian lemmatizer at the CONLL 2018 Shared Task achieved the accuracy of 96.57 %,[2] leaving little room for improvement. However, we propose that it is still possible to reduce the errors even further by utilizing existing linguistic resources such as rule-based systems and lexicons. Our proposal enables to unite the strengths of both approaches—the neural representation learning and the symbolic rules of otherwise hard to predict words.

VABAMORF [3] is a rule-based Estonian morphological analyser that for each word generates all possible morphological analyses consisting of lemma, part-of-speech and morphological features. We propose to integrate the lemmas generated by the

[1]Corresponding Author: Kirill Milintsevich; E-mail: kirill.milintsevich@ut.ee
[2]`https://universaldependencies.org/conll18/results-lemmas.html`

VABAMORF analyser directly into the neural lemmatizer, thus enabling the model to rely on both sources of information—the regularities learned by the neural model and the candidates proposed by the analyser. Our model encodes both the input word and the lemma candidates generated by VABAMORF, and passes both representations into a decoder. The decoder then learns to benefit from the second input by passing it through an additional layer of attention.

We conduct experiments on the Estonian Universal Dependencies (UD) dataset and show that our model with additional VABAMORF inputs achieves significantly higher results compared to the baseline model trained only on the UD training set. Moreover, our VABAMORF enhanced model also surpasses the best Estonian lemmatization result of the CoNLL 2018 Shared Task.

2. Previous Work on Estonian Lemmatization

The previous work on Estonian lemmatization derives from two sources. The first is the rule-based VABAMORF morphological analyzer [3] that in addition to POS tags and morphological features also produces lemmas. The lemmatization module is based on a lexicon which, according to Kaalep and Vaino [3], is estimated to cover ca. 97 % of tokens in any Estonian text. The system also has a guesser module that attempts to generate lemmas for unknown words. Although VABAMORF also features a statistical disambiguator, approximately 13.5 % of all words are expected to remain ambiguous for various reasons [3]. Some of these ambiguities are solved by considering the wider textual context [4]. According to our knowledge, there is only one previous work that has evaluated the performance of VABAMORF lemmatizer on common benchmark UD datasets [5]. According to Leman [5], the lemmatization accuracy of the VABAMORF system on the UD v2.3 test set is about 95.2 %.

The second line of work involving lemmatizing Estonian originates from the CoNLL 2017 and 2018 Shared Tasks [6, 7] and the SIGMORPHON 2019 Shared Task [8]. The most widely known systems from these competitions are the Stanford Stanza [9], UdPipe [10] and TurkuNLP [11] neural lemmatizers. These systems also exemplify the two main approaches used in neural lemmatization systems. Both Stanza and TurkuNLP are based on the sequence-to-sequence architecture, where the lemma for a word is generated character by character. The UDPipe model, on the other hand, utilizes a classification approach. Based on training set, a set of rules for transforming a word into its lemma are extracted. On the current Estonian UD v2.5 test set, the lemmatization results are 96.05 % for Stanza[3] and 90.6 % for UDPipe.[4] The TurkuNLP achieved the best performance on Estonian UDv2.2 dataset in the CoNLL-2018 Shared Task with 96.57 %. However, it is unknown how well it performs on the UD v2.5 test set.

3. Lexicon-Enhanced Lemmatization Model

The core of our model is the Stanza lemmatizer [9] which is a sequence-to-sequence encoder-decoder model. Stanza takes the character-level word representation and the

[3] https://stanfordnlp.github.io/stanza/performance.html
[4] http://ufal.mff.cuni.cz/udpipe/models

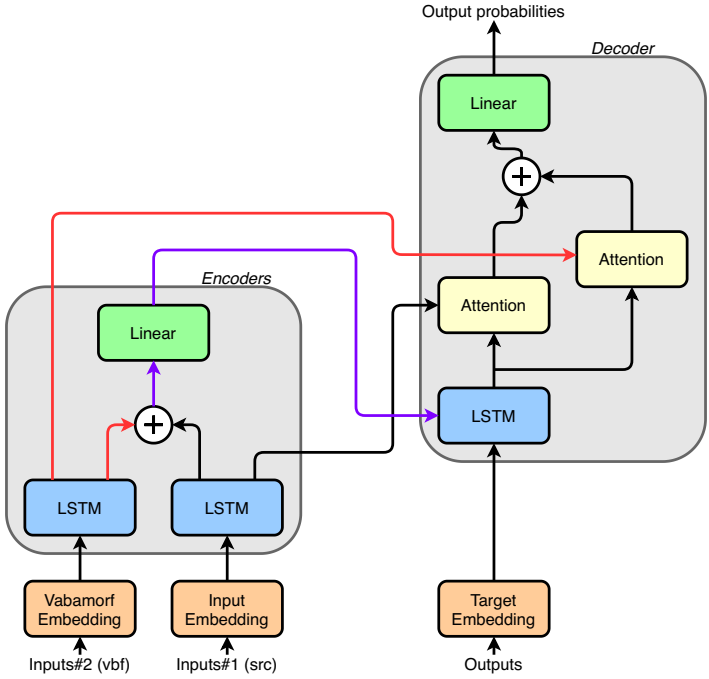

Figure 1. Dual encoder architecture for the lexicon-enhanced lemmatizer

POS tag embedding as input and processes them with a bidirectional LSTM encoder. Then, it passes the encoder outputs to a LSTM decoder. The decoder applies a soft dot attention layer after every LSTM step. Finally, the output is constructed with the greedy search over the decoder outputs.

The overall architecture of our model is presented on **Figure 1**, which shows the changes made to the Stanza lemmatizer. In particular, we add another encoder that takes the candidates generated by the VABAMORF analyser as input. The outputs of both encoders are combined with a linear layer and fed to the decoder. Moreover, we add another attention layer to the decoder that attends to the VABAMORF candidates encoded by the second encoder. This helps the model to better choose the appropriate features from both encoders. The outputs of both attention layers are finally combined with a linear layer.

Finally, in addition to POS tag, we also add morphological features to the input of the first encoder. We will show that it improves the lemmatization for Estonian and can potentially be helpful for other highly inflectional languages as well.

4. Experimental Setup

For training and testing the models we use the Estonian UD v2.5 treebank [12]. The treebank is in the CoNLL-U format and conforms to the Universal Dependencies project annotation standards [13]. It contains 437 769 tokens annotated with POS tags, morphological features, lemmas, and syntactic information. The treebank is based on the Estonian Dependency Treebank (EDT), created at the University of Tartu.

Table 1. Experimental results for our Vabamorf-enhanced model, the baseline with the empty second encode (Non-Enhanced), and the Stanza baseline. The OOV column shows the accuracy on the out-of-vocabulary words only

Rank	Model	Accuracy	
		All words	OOV
1	LEXENLEM (Vabamorf-enhanced)	96.87 ± 0.17	88.64
2	LEXENLEM (Non-Enhanced)	96.36 ± 0.20	86.16
3	Stanza	96.11 ± 0.20	83.86

All models were evaluated on the Estonian UD test set with POS tags and morphological features predicted by the Stanza pipeline [9] that achieved 94.54 F1-score for all tags (upos, xpos, and feats). The predictions for each run were ranked and tested with paired bootstrap resampling [14]. Each score is accompanied by 95 % confidence intervals obtained with 10,000 resamples. The p-value shows if the difference with the next system is statistically significant. If the p-value < 0.05, we rank the system higher than the following one. As baseline, we use the default Stanza model for Estonian that has been trained on the same UD v2.5 dataset.

5. Results

We conducted experiments with our LEXicon-ENhanced LEMmatization (LEXENLEM) model in two different settings that differ in the input to the second encoder. In the baseline version (LEXENLEM Non-Enhanced), the second encoder receives no input. The Vabamorf-Enhanced version receives via the second encoder all distinct lemma candidates generated by the VABAMORF morphological analyser. If there are several lemmas then they are simply concatenated.

As can be seen from the **Table 1**, the Stanza baseline model was ranked the least. Our Non-Enhanced model providing better results can be explained by the addition of morphological features to the input of the first encoder. The enhanced model with the second encoder utilizing the Vabamorf predictions outperforms both baselines. According to the bootstrap test, the differences between all models are statistically significant at the level of $p < 0.05$. The difference between the Vabamorf-enhanced LEXENLEM and the Stanza baseline are especially visible when inspecting the accuracy of the out-of-vocabulary words. Overall, our Vabamorf-Enhanced lemmatization model achieves a new best result on the Estonian UD test set.

5.1. The Effect of Word Formation Symbols

Lemmas in Estonian EDT are additionally annotated with the word formation information, specifically compounding and morphological derivation. For example, the lemma for the word *ostusedelisse* (in the shopping list) is *ostu_sedel*. The underscore in the lemma shows that the word is compound and in fact consists of two words: *ostu* (shopping) and *sedel* (a list).

We analyzed the errors made by all models and found that indeed, many errors are related to the misplacement of the word formation symbols. **Table 2** shows the distribution

Table 2. Division of different types of errors made by the models

Model	Missing		Misplaced		Misc	Total
	COM	DER	COM	DER		
LEXENLEM (Vabamorf)	82	225	197	124	890	1518
LEXENLEM (Non-Enhanced)	107	211	166	120	1163	1767
Stanza	146	192	150	82	1316	1886

of the errors. In the table, **COM** signifies the symbol "_" separating the compound parts in a compound word, **DER** denotes the the derivational symbols "+" and "=". **Missing COM** and **Missing DER** stand for the errors when the respective symbol is present in the gold lemma but not present in the predicted lemma and if removed, the prediction is correct (e.g. correct: *"laua_naaber"*; predicted: *"lauanaaber"*). **Misplaced COM** and **Misplaced DER** stand for the errors when the respective symbol is present in both gold and predicted lemmas but is misplaced in the predicted (e.g. correct: *"ostu_sedel"*; predicted: *"ostus_edel"*) or it is not present in the gold but is present in the predicted and if removed, the prediction is correct (e.g. correct: *"seostamine"*; predicted: *"seosta=mine"*). **Misc** stands for all the other errors. As it can be seen, the number of errors related to word formation annotation symbols is roughly the same across all models, while the best performing model reduces the number of MISC errors.

Thus, we created another version of the data where all word formation symbols were removed, and trained our models on this modified dataset. **Table 3** shows the results of these experiments. The Stanza baseline results are obtained by removing the word formation symbols before evaluation. All the models retain their rankings and show improvement in accuracy for more than 1 % for all words and more than 5.5 % for out-of-vocabulary words.

Table 3. Experimental results for our Vabamorf-enhanced model, the baseline with the empty second encode (Non-Enhanced), and the Stanza baseline without word formation symbols. The OOV column shows the accuracy on the out-of-vocabulary words only

Rank	Model	Accuracy	
		All words	OOV
1	LEXENLEM (Vabamorf-enhanced)	98.11 ± 0.13	94.14
2	LEXENLEM (Non-Enhanced)	97.66 ± 0.15	91.88
3	Stanza	97.29 ± 0.15	89.99

5.2. The Effect of the Vabamorf Settings

VABAMORF in its basic setting is a morphological analyser that for each word returns all possible morphological analyses, including lemmas. In addition to this basic setting, there are several additional modules that can be applied. The disambiguator module uses a statistical HMM model to select the most likely analysis for each word in context. The proper name module attempts to recognise proper names. Finally, the guesser module attempts to guess the lemmas for words that are not present in the system's dictionaries.

Table 4. Results of the VABAMORF settings ablation experiments. Basic is the VABAMORF system without additional modules; +PN adds the proper name module, +DIS adds the disambiguation module and +Guesser adds the guesser module

Setting	Dev Accuracy	Test Accuracy	Test OOV
Basic	**99.14**	**98.11**	**94.14**
+PN	99.09	98.04	93.82
+DIS	99.04	97.99	93.79
+PN +DIS	99.05	97.95	93.46
+PN +DIS +Guesser	99.05	97.95	93.46

The application of all these modules can have an effect on our lexicon-enhanced model as well. Thus, we performed a set of ablation experiments with the different settings of the Vabamorf system. All these experiments were done on the dataset with the word formation symbols removed.

Table 4 presents the results on both development and test set. On both evaluation sets, the basic VABAMORF without additional modules performs the best, suggesting that giving more ambiguous input to the second encoder improves our model. This may be because increasing the ambiguity of the morphological analyses raises the likelihood that the additional input includes the correct lemma. When the guesser module is turned on, VABAMORF attempts to predict the lemma for unknown words using an inferior algorithm while without the guesser it skips unknown words and thus leaves the prediction task to our model.

5.3. The Effect of the Vabamorf Candidates

The central idea of the proposed approach is to use an external system or lexicon to influence the lemmatization model towards correct predictions if the external systems knows the correct lemma. To analyze if our model succeeded in that, we analyzed the errors made by the Vabamorf-Enhanced model and the baseline Non-Enhanced model that did not receive any input into the second encoder. **Figure 2** demonstrates the effect of the VABAMORF candidates on the model's performance. As can be seen from the graph, in the majority of the cases when the Vabamorf-enhanced model predicts the correct lemma and the Not-Enhanced model's prediction is wrong (the first column), at least one of the candidates passed to the Vabamorf-enhanced model is correct, suggesting that the additional input indeed influenced our model towards making a correct prediction.

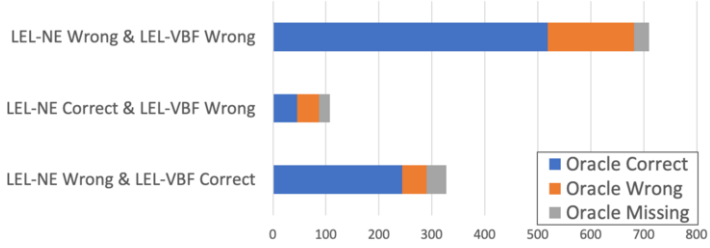

Figure 2. Comparison of the errors made by the**L**EX**E**N**L**EM **N**ot-**E**nhanced (LEL-NE) and the **L**EX**E**N**L**EM **Va**mor**f**-enhanced (LEL-VBF) models. Oracle stands for all the candidates produced by VABAMORF

The third column shows the number of cases when both models predicted wrong lemmas even though the Vabamorf-enhanced model received the correct candidate during the inference. This was mostly caused by the wrong capitalization of loan words (e.g. Jazz, Rock), adding -i ending to the foreign names (e.g. *Johni, *Pauli, *Kölni), and wrong disambiguation of *see* and *tema* which have the same plural form *neid*. Another reason for this type of errors is incorrectly predicted POS and/or morphological tags. For example, the form *praeme_Verb.Present.1per.Plur* was wrongly tagged as **praeme_Noun.Part.Plur* and thus lemmatized as **praed* while the correct lemma would be the infinitive *praadima*. This can signify that the information encoded in the morphological tags has more weight than the additional lemma candidates provided to the model.

There are also a small number of cases where the Non-Enhanced model predicts correctly while the Vabamorf-Enhanced model generates a wrong lemma (the middle column), even though there is a correct lemma in the VABAMORF candidates. These cases are also mostly related to the erroneous POS and morphological tags.

6. Conclusion

We presented a novel approach to Estonian lemmatization that enables a seq2seq encoder-decoder model to benefit from the external VABAMORF morphological analyser. Our hybrid model achieves a new state-of-the-art results in lemmatization on the Estonian UD test set with 96.87 % when the word formation symbols are considered and 98.11 % with word formation symbols removed. We also analyzed the error patterns of our model and found that many errors are related to incorrect POS and morphological predictions which influence the model to generate incorrect lemmas even when the correct lemma candidate is provided. This suggests that in order for the Vabamorf-enhanced model to fully gain from the extra input, it might be beneficial to downweight the relative importance of the POS and morphological info when the external candidates are provided.

References

[1] Kanis J, Skorkovská L. Comparison of different lemmatization approaches through the means of information retrieval performance. In: Proceedings of TSD 2010; 2010. p. 93–100.

[2] Konkol M, Konopík M. Named entity recognition for highly inflectional languages: effects of various lemmatization and stemming approaches. In: Proceedings of TSD 2014; 2014. p. 267–274.

[3] Kaalep HJ, Vaino T. Complete morphological analysis in the linguist's toolbox. Congressus Nonus Internationalis Fenno-Ugristarum Pars V. 2001:9–16.

[4] Kaalep HJ, Kirt R, Muischnek K. A trivial method for choosing the right lemma. In: Proceedings of Baltic HLT 2012. vol. 247; 2012. .

[5] Leman LK. Comparative analysis of neural Network Based Lemmatizers in the Estonian Language [Bachelor's Thesis]. University of Tartu; 2019.

[6] Zeman D, Popel M, Straka M, Hajič J, et al. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In: Proceedings of the CoNLL 2017 Shared Task; 2017. p. 1–19.

[7] Zeman D, Hajic J, Popel M, Potthast M, Straka M, Ginter F, et al. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In: Proceedings of the CoNLL 2018 Shared Task; 2018. p. 1–21.

[8] McCarthy AD, Vylomova E, Wu S, Malaviya C, Wolf-Sonkin L, Nicolai G, et al. The SIGMORPHON 2019 Shared Task: Morphological Analysis in Context and Cross-Lingual Transfer for Inflection. In: Proceedings of the SIGMORPHON; 2019. p. 229–244.

[9] Qi P, Zhang Y, Zhang Y, Bolton J, Manning CD. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. arXiv preprint arXiv:200307082. 2020.

[10] Straka M, Hajic J, Straková J. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In: Proceedings of LREC'16; 2016. p. 4290–4297.

[11] Kanerva J, Ginter F, Miekka N, Leino A, Salakoski T. Turku neural parser pipeline: An end-to-end system for the CoNLL 2018 Shared Task. In: Proceedings of the CoNLL 2018 Shared Task; 2018. p. 133–142.

[12] Muischnek K, Müürisep K, Puolakainen T, Aedmaa E, Kirt R, Särg D. Estonian dependency treebank and its annotation scheme. In: Proceedings of TLT'13; 2014. p. 285–291.

[13] Zeman D, Nivre J, Abrams M, et al. Universal Dependencies 2.5; 2019. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. Available from: `http://hdl.handle.net/11234/1-3105`.

[14] Koehn P. Statistical significance tests for machine translation evaluation. In: Proceedings of EMNLP'04; 2004. p. 388–395.