PoS-tagging and lemmatization with a deep recurrent neural network

Gábor Ugray

memoQ Translation Technologies
gabor.ugray@memoq.com

Abstract. Neural networks have been shown to successfully solve many natural language processing tasks previously tackled by rule-based and statistical approaches. We present a deep recurrent network with long short-term memory, identical to engines used in machine translation, to solve the problem of joint PoS-tagging and lemmatization in Hungarian and German. Our model achieves comparable or superior results to a state-of-the-art statistical PoS tagger. We are able to enhance the Hungarian model's performance, as measured on a manually annotated sample unrelated to the initial training corpus, through an additional synthesized dataset.

Keywords: PoS-tagging, lemmatization, neural networks, LSTM, Hungarian, German

1 Introduction

In recent years we have seen deep neural networks applied to many linguistic modeling tasks that were previously tackled by statistical or rule-based approaches. Németh & Ács [1] achieved promising results for Hungarian hyphenation. Chinese word segmentation is a challenge because of the scripts's lack of spaces. Zheng, Cheng & Xu [2] have shown that a neural model yields results competitive with the state of the art in word segmentation and PoS-tagging.

While many of these approaches formulate the problem as a classification task, Rei, Crichton & Pyysalo [3] have studied sequence labeling and found that an attention model improves performance.

In the problem domain of morphologically rich languages, Yildiz & al [4] have trained neural networks to disambiguate the output of a rule-based morphological analyizer (MA). Zalmout & Habash [5] have successfully used the same approach for Arabic.

In the present paper, we set out to explore a related, but slightly broader, problem: joint PoS-tagging and lemmatization. We define the challenge as a sequence-to-sequence transformation identical to machine translation (MT) between different natural languages. We train an off-the-shelf neural MT engine and achieve outcomes that are competitive or superior to a state-of-the-art PoS tagger. We show that we can boost the neural model's performance on new domains through a training dataset

synthesized via the state-of-the-art statistical PoS tagger trained on a relatively small, manually annotated corpus.

2 Experimental setup

2.1 Recurrent network with long short-term memory and attention

We formulate the joint task of PoS-tagging and lemmatization as a sequence-to-sequence transformation [6]. The transformation's input is the token to be tagged and lemmatized, surrounded by a chosen number of preceding and following tokens for context. The output is the lemma, followed by one or more tags. For illustration, *Table 1* shows the first few input-output pairs generated from a tokenized sentence, with a context window of 5 surface tokens.

[Beg] Néhány [End] pillanat múl■ tán hangot hallott az	néhány [/Num] [Nom]
Néhány [Beg] pillanat [End] múl■ tán hangot hallott az ember	pillanat [/N] [Nom]
Néhány pillanat [Beg] múl■ tán [End] hangot hallott az ember ,	múl■ tán [/Post]
Néhány pillanat múl■ tán [Beg] hangot [End] hallott az ember , amely	hang [/N] [Acc]
Néhány pillanat múl■ tán hangot [Beg] hallott [End] az ember , amely a	hall [/V] [Pst.NDef.3Sg]
Néhány pillanat múl■ tán hangot hallott [Beg] az [End] ember , amely a táb■ lák	az [/Det]
pillanat múl■ tán hangot hallott az [Beg] ember [End] , amely a táb■ lák mög■ ül	ember [/N] [Nom]

Table 1: Sample input and output sequences from the neural model's training corpus.

Since we fix the context window's size in surface tokens, a convolutional neural network (CNN) might at first seem like a more natural choice. The experience of neural machine translation, however, is that decomposing the input into subword tokens is a successful way to address the open vocabulary problem. In our model, therefore, we further tokenize both the input tokens and the the target lemma using byte-pair encoding (BPE) [7]. The result is a dataset with random-length sequential input and output.

The models we train for the various experiments are identical, off-the-shelf neural MT models using a bidirectional LSTM and attention [8]. We use OpenNMT's [9] default parameters: 2 hidden layers with 500 hiden units. All models are trained for 13 epochs, with SGD optimization and a learning rate decaying from 1.0 by a factor of 0.7 from epoch 9 onwards. Word embeddings have 500 dimensions.

We use a shared BPE model for the source (surface words) and the target (lemmata), with 12.5 thousand merges. This is a comparatively small vocabulary for neural MT models. Our aim, however, is to model words, not sentences, so we feel even a smaller choice might be warranted. The begin/end delimiters in the source, and the morphological tags in the target, are preserved as distinct vocabulary words; they are exempt from BPE segmentation.

2.2 Experiments

We devise a set of experiments to answer various exploratory questions about the neural approach.

Direct comparison: Hungarian. How does the accuracy of a neural model compare with a state-of-the-art tagger, when trained and evaluated on a 19:1 split of the same annotated corpus? We train both PurePos [10] [11] and a neural model on 95% of the Szeged corpus [12], and measure accuracy on a 5% evaluation set, after a random split.

This experiment is also a replication study, because for the comparison we remeasure PurePos's reported tagging and lemmatization accuracy. We perform an initial measurement without a morphological analyzer (MA). PurePos's best numbers, however, were reported with an integrated MA. We therefore also reproduce that outcome using the recently open-sourced emMorph analyzer [13], in conjunction with a version of the Szeged corpus converted to the emMorph/HuMor formalism.

Direct comparison: German. The publications related to PurePos that we are aware of are all based on Hungarian datasets, but we are curious how well its results generalize to other languages. We therefore perform the same measurements using a comparable German annotated corpus, Tiger [14]. In this case, there is no compatible MA to include. In addition to PurePos, we also measure the tagging accuracy of NLTK's classifier-based tagger.

Synthesized training data. Can we improve the neural model by synthesizing additional training data? For our particular supervised learning scenario, the amount of manually annotated text is limited. Meanwhile, PurePos can generalize well to new input in part because of the integrated MA. We first train PurePos on the Szeged corpus, then use it to tag and lemmatize a different, 923-thousand-segment corpus. This automatically annotated dataset, together with the original Szeged corpus's training set, is used to train a neural model.

We compare this neural model's performance with PurePos on the Szeged corpus's validation set, and on a small manually annotated evaluation dataset. The aim is to test whether the neural model can learn a meaningful amount of Hungarian morphology from the examples transmitted through the larger synthesized traininig corpus.

3 Data and preparation

Dataset	Sentences	Tokens	Types	Full tags	Tag vocab
Szeged	81,967	1,485,306	152,057	1,246	169
Tiger	50,472	888,238	89,383	694	78
Szeged+Synth	1,005,464	10,330,582	609,359	4,763	214
Eval	491	4,959	2,331	264	120

Table 2: Key statistics about the datasets used for the experiments.

3.1 The Szeged corpus and Tiger

We used a version of the Szeged corpus where the annotations have been converted to the formalism of HuMor/emMorph[1]. The numbers related to the Szeged corpus in *Table 2* are from this converted version.

In HuMor's output, a sequence of tags encodes each word's morphological information. E.g., [/N][Pl][Acc] is a noun, in plural form and with an accusative case marker. The *Full tags* column in *Table 2* refers to the number of distinct tag sequences attested in the data; *Tag vocab* is the number of distinct bracketed tags. Referring back to *Table 1*, we can see that in the neural model's training data we chose to treat each bracketed tag as a separate token. This results in a smaller vocabulary and the possibility that the model can output even rare (but correct) sequences not attested in the training data.

The Tiger corpus uses a small set of part-of-speech categories and has additional annotations for each word's inflectional categories. As an example, a particular instance of "größte" is lemmatized as "groß"; the PoS label is "ADJA"; and the inflectional categories are "case=acc|number=sg|gender=fem|degree=sup". For our purposes, we convert this to the following sequence of bracketed tags:

[ADJA][case=acc][number=sg][gender=fem][degree=sup]

3.2 Incompatible annotations in the Szeged corpus

As we shall see in the Results section, PurePos's tagging accuracy fell from 97.55% to 79.72%, and its lemmatization accuracy from 96.38% to 90.28%, when we first ran it with an MA, as opposed to relying only on the built-in guesser. This clearly indicated an incompatibility between the converted corpus annotations and emMorph's actual output.

We extracted words where emMorph's analyses did not include the annotation in the corpus. The problem was severe: it affected 32 thousand of the corpus's 152 thousand types, and 312 thousand of its 1.4 million tokens. Because it is not feasible to alter emMorph's rules and lexical database, we chose to adjust the corpus's annotations to make them compatible with emMorph's observed output.

Some problems were trivial, e.g., a difference in the way some punctuation marks were labeled. We also observed that the information following the pipe character ("|") was often incompatible, e.g., emMorph's analysis including a marker about the Latin origin of some words, which is not part of the corpus's annotations. We chose to remove everything from the first pipe onwards in every bracketed tag, both in the corpus and in emMorph's output.

Finally, there was a large number of words where all of emMorph's analyses included at least one derivational suffix, while the corpus annotation was the fully derived form. E.g., "földrajzos" is annotated as "földrajzos[/Adj][Nom]" in the corpus, but analyzed only as "földrajz[/N][_Adjz:s/Adj][Nom]" or

[1] The converted corpus was kindly provided by Veronika Vincze. Unfortunately, we haven't been able to obtain published information about the conversion process.

"földrajz[/N][_Nz:s/N][Nom]" by emMorph. We managed to identify a handful of such patterns and replaced the corpus annotation with the closest, slighly less derived analysis from emMorph.

We did not aim for perfection, as the pattern matching effort soon began to yield diminishing returns. We stopped when we reduced the discrepancy to 7,275 types with 24,152 token instances. With this effort, PurePos's tagging accuracy no longer deteriorated with the MA enabled, and its lemmatization accuracy increased slightly. Details are included in the Results section.

Making PurePos work with morphology was critical for the key experiment, which involves the automatic PoS-tagging and lemmatization of a large dataset with many types and lemmas not attested in the Szeged corpus.

3.3 Synthesized dataset

For the synthesized training data we used 923 thousand segments from open sources[2]. The corpus consists of 5% JRC-Acquis, 7% Europarl, 9% modern literature, and 79% movie subtitles. This particular corpus was chosen because it is sufficiently versatile; we had originally created it as a bilingual dataset for training a machine translation engine. For this research's purposes, we took a random subset of the original bilingual dataset's Hungarian sentences.

To prepare for tagging, we tokenized the already sentence-segmented corpus using quntoken, the standalone version of the e-magyar toolchain's [15] emToken component.

We did not find a trivial way to use emMorph as an integrated MA directly invoked by PurePos. We therefore first extracted all surface forms (types), executed HFST from the command line, and fed the analyses via PurePos's morphology table option. For this, we needed to slightly alter PurePos's source code, whose published version ignores lemmata from the morphology table and only returns tags.

Executing HFST itself posed a small challenge. Analyzing the 600 thousand extracted surface forms took over 12 hours, and was only possible in a dozen smaller batches. On larger batches the tool predictably runs out of memory and crashes before completing its job, even with the 1-second timeout option.

3.4 Manually annotated evaluation set

After training a neural model on an automatically tagged corpus, there are multiple ways to evaluate it.

First, we can measure to what extent it coincides with PurePos on a smaller, randomly selected validation set. This, however, would not measure how well the neural system learns to model linguistic reality; it would only show how well it learns to replicate PurePos's model. Second, we can check whether the neural model trained on the large corpus makes better predictions on the Szeged corpus's 5% validation set.

[2] http://opus.nlpl.eu/

The most insightful evaluation, however, is on a manually annotated gold standard that is not part of the Szeged corpus. This approach allows us to compare the neural model's performance to PurePos in a new domain.

To create the evaluation set we separated a small random sample of the synthesized corpus and manually corrected its annotations. This 492-sentence evaluation set was excluded from the neural model's training material. For the manual review we relied on the output of PurePos and emMorph's analyses, and frequently cross-checked with the Szeged corpus to mirror its conventions as closely as possible.

We share the manually annotated evaluation dataset, along with the output of the different models, as an Excel file[3].

3.5 Limitations

In addition to the remaining inconsistency in the Szeged corpus's annotations, we acknowledge a further limitation of our experimental setup. The 19:1 split of the corpus is different from the standard 9:1 split, and all of our experiments were done with a single random split. For more reliable results, a full roll would be required, retraining models repeatedly and alternating through different subsets of the corpus for evaluation. Due to limited time and resources, this was unfortunately not possible.

4 Results

4.1 Evaluation on the Szeged corpus

The initial question we set out to answer is whether a neural model can achieve comparable accuracy, or potentially even outperform a state-of-the-art tagger, as measured on a 19:1 split of the annotated Szeged corpus. *Table 3* shows the results we obtained with the converted corpus.

The *Tag-Full* column is tagging accuracy, as measured by the entire tag sequence, and counted by tokens. *Tag-First* is more permissive: it only checks the first bracketed tag (typically, although not always, the part of speech). *Lemma-Strict* is lemmatization accuracy; *Lemma-CI* is a more permissive, case-insensitive measure.

Model	Tag-Full	Tag-First	Lemma-Strict	Lemma-CI
PurePos	97.55%	98.58%	96.38%	96.99%
PurePos+MA	79.72%	81.24%	90.28%	91.53%
Neural	97.99%	98.79%	98.86%	98.95%

Table 3: Accuracy of the different taggers on the 5% validation set of the converted Szeged corpus.

[3] https://jealousmarkup.xyz/files/MSZNY2019-PoS-EvalSet.xlsx

In this setup, the neural model outperforms PurePos without an MA. As discussed in the previous section, adding an MA produced drastically bad results because of the incompatibility between emMorph's output and the corpus's annotations. Therefore, in *Table 4* we present the results of the same experiment, but this time repeated on the corpus with the adjusted annotations.

Model	Tag-Full	Tag-First	Lemma-Strict	Lemma-CI
PurePos+MA	97.41%	98.57%	97.24%	97.69%
Neural (Szeged)	97.89%	98.83%	98.51%	98.70%
Neural (Szeged+ Synth)	98.01%	98.88%	98.74%	98.96%

Table 4: Accuracy of the different taggers on the 5% validation set of the converted and adjusted Szeged corpus.

PurePos's tag accuracy with an MA is now effectively identical to its accuracy without an MA from the previous experiment; its lemmatization accuracy has improved. We would expect an improvement across the board if the corpus annotations had been completely brought in line with emMorph.

The neural model, again, slightly outperforms PurePos when trained on the same corpus. The model that was trained on the extended corpus (including Szeged's training set plus the 923-thousand-segment synthesized dataset) yields additional improvements. This is interesting, because the improvements are detected on Szeged's validation set, while the synthesized training data is based on an entirely different corpus.

4.2 Evaluation on Tiger

Table 5 presents the results from Tiger, the 888-thousand-word German annotated corpus, after a 19:1 training/evaluation split. The first row, NLTK-CB, shows the tagging accuracy of the NLTK toolkit's classifier-based tagger. That tagger does not perform lemmatization, and only produces a single tag per token, so the other metrics are not applicable.

Model	Tag-Full	Tag-First	Lemma-Strict	Lemma-CI
NLTK-CB	n/a	94.07%	n/a	n/a
PurePos	84.82%	97.19%	96.57%	97.10%
Neural	91.85%	98.01%	98.43%	98.58%

Table 5: Accuracy of the different taggers on the 5% validation set of the German Tiger corpus.

PurePos outperforms the classifier-based tagger, and the neural model outperforms PurePos on all metrics. The most drastic difference is in the full tagging accuracy. We conjecture that this may be related to the neural model's 5-word window, which is in a sense larger than PurePos's third-order Hidden Markov Model. We suspect that the

correct value of German inflectional categories (e.g., the gender and number of a form like "größte") might be driven by constituents farther away in the sentence. We did not, however, test this conjecture.

4.3 Annotated test set

The key experiment was the evaluation of the different models on a manually annotated dataset. *Table 6* shows the results.

Model	Tag-Full	Tag-First	Lemma-Strict	Lemma-CI
PurePos	95.72%	97.90%	95.62%	96.51%
PurePos+MA	96.87%	98.21%	97.06%	97.90%
Neural (Szeged)	93.83%	96.53%	94.98%	97.70%
Neural (Szeged+ Synth)	96.55%	97.98%	96.85%	97.70%

Table 6: Accuracy of the different taggers on the small, manual annotated gold standard dataset.

Unsurprisingly, PurePos with an MA outperforms PurePos without morphology. Obviously, both PurePos models were trained on the Szeged corpus's 95% training set, there being no other ground truth. "Neural (Szeged)" is the neural model trained on the same corpus. It significantly underperforms PurePos, particularly on the strict metrics.

"Neural (Szeged+Synth)" is the model that we trained on the extended corpus. On the manual evaluation set it fails to reach PurePos's accuracy with morphology, but it does outperform PurePos without an MA. In particular there is a big improvement in terms of full tagging accuracy and strict lemmatization accuracy.

4.4 A qualitative look

The filters of the accompanying Excel file with the results of each model on the evaluation set allow for a lot of exploration. Where "Neural (Szeged)" gets lemmata wrong we frequently see missing morphological insight, which is then corrected in "Neural (Szeged+Synth)". One example would be "odalbben" as the lemma returned for "odalent". Because the neural sequence-to-sequence system's output is generated recursively from the network's activation state, the model always produces *some* output, and that output can easily contain sequences that were never attested in the training data, or which simply don't make much sense.

We also see a few of the sort of "hallucinations" that have been observed in neural MT systems, but which are unimaginable in rule-based tools. One example would be "Robert" as the lemma returned for "4)".

115 tokens in the evaluation set are out-of-vocabulary (OOV), i.e., they were not attested in the training data. For 91 of these, the neural model returns a correct lemma, which we see as evidence that the model has acquired morphological insight.

5 Conclusion

We have shown that a deep neural network with a bidirectional LSTM topology can learn to jointly lemmatize and PoS-tag text in dissimilar languages such as Hungarian and German. Neural models achieve comparable or superior results to state-of-the-art statistical PoS taggers such as PurePos, even when these incorporate a morphological analyzer. When trained on the relatively small manually annotated corpora that are available for the PoS-tagging task, the neural model has difficulty generalizing to a new domain. However, if we boost the neural model with a large synthetic dataset automatically annotated by a traditional morphology-aware PoS-tagger, it achieves comparable results on a new domain as well.

We achieved these results using an off-the-shelf neural MT engine without any parameter tuning. We are confident that the results can be improved significantly by exploring different network dimensions and optimization methods, different context windows, and more or less aggressive sub-word segmentation. Much larger automatically annotated datasets are also easy to create, promising to broaden the neural model's morphological coverage even further.

Perhaps most importantly, for supervised learning tasks such as PoS-tagging, the core training data's amount and quality has a tremendous impact on the outcome.

References

1. Gergely Dániel Németh, Judit Ács: Hyphenation using deep neural networks. *In V. Vincze (szerk.) XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018)*. Szegedi Tudományegyetem, Szeged. (2018) pp. 146-158.
2. Xiaoqing Zheng, Hanyang Chen, Tianyu Xu: Deep Learning for Chinese Word Segmentation and POS Tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 647–657. (2013) Seattle, WA, USA
3. Marek Rei, Gamal K.O. Crichton, Sampo Pyysalo: Attending to Characters in Neural Sequence Labeling Models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 309–318, Osaka, Japan (2016)
4. Eray Yildiz, Caglar Tirkaz, H. Bahadir Sahin, Mustafa Tolga Eren, Omer Ozan Sonmez: A Morphology-Aware Network for Morphological Disambiguation. In *Proceedings of AAAI. AAAI Press*, pp. 2863–2869. (2016)
5. Nasser Zalmout, Nizar Habash: Don't Throw Those Morphological Analyzers Away Just Yet: Neural Morphological Disambiguation for Arabic. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 704-713. ACL, Copenhagen, 2017.
6. Ilya Sutskever, Oriol Vinyals, Quoc V. Le: Sequence to Sequence Learning with Neural Networks. In *NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems*. Vol. 2., pp. 3104-3112, Montreal, Canada (2014)
7 Sennrich, Rico, Haddow, Barry and Birch, Alexandra: Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*. Berlin, Germany.
8 Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio: Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations* (2016)

9 Klein, Guillaume; Kim, Yoon; Deng, Yuntian; Crego, Josep; Senellart, Jean; Rush, Alexander M.: OpenNMT: Open-source Toolkit for Neural Machine Translation. In *Proceedings of ACL* 2017.

10 G. Orosz, A. Novák: PurePos 2.0: a hybrid tool for morphological disambiguation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013)*, pp. 539–545, Hissar, Bulgaria, 2013. INCOMA Ltd. Shoumen, BULGARIA.

11 G. Orosz, A. Novák: PurePos – an open source morphological disambiguator. In *B. Sharp, M. Zock (eds.): Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science*, pp. 53–63, Wroclaw, 2012.

12 D. Csendes, J. Csirik, T. Gyimóthy: The Szeged corpus: a POS tagged and syntactically annotated hungarian natural language corpus. In *Sojka, P., Kopecek, I., Pala, K. (eds.) TSD 2004. LNCS (LNAI), vol. 3206*, pp. 41–47. Springer, Heidelberg (2004)

13 Attila Novák; Borbála Siklósi; Csaba Oravecz (2016): A New Integrated Open-source Morphological Analyzer for Hungarian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Portorož, pp. 1315–1322.

14 Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, Hans Uszkoreit: TIGER: Linguistic Interpretation of a German Corpus. In Journal of Language and Computation, 2004 (2), 597-620.

15 Váradi T., Simon E., Sass B., Gerőcs M., Mittelholcz I., Novák A., Indig B., Prószéky G., Farkas R., Vincze V.: Az e-magyar digitális nyelvfeldolgozó rendszer. Magyar Számítógépes Nyelvészeti Konferencia (2017)