Kamil Ekštein (Ed.)

LNAI 11697

Text, Speech, and Dialogue

22nd International Conference, TSD 2019
Ljubljana, Slovenia, September 11–13, 2019
Proceedings

Czech Text Processing with Contextual Embeddings: POS Tagging, Lemmatization, Parsing and NER

Milan Straka, Jana Straková, and Jan Hajič

Charles University, Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics
http://ufal.mff.cuni.cz
{straka,strakova,hajic}@ufal.mff.cuni.cz

Abstract. Contextualized embeddings, which capture appropriate word meaning depending on context, have recently been proposed. We evaluate two methods for precomputing such embeddings, BERT and Flair, on four Czech text processing tasks: part-of-speech (POS) tagging, lemmatization, dependency parsing and named entity recognition (NER). The first three tasks, POS tagging, lemmatization and dependency parsing, are evaluated on two corpora: the Prague Dependency Treebank 3.5 and the Universal Dependencies 2.3. The named entity recognition (NER) is evaluated on the Czech Named Entity Corpus 1.1 and 2.0. We report state-of-the-art results for the above mentioned tasks and corpora.

Keywords: contextualized embeddings, BERT, Flair, POS tagging, lemmatization, dependency parsing, named entity recognition, Czech

1 Introduction

Recently, a novel way of computing word embeddings has been proposed. Instead of computing one word embedding for each word which sums over all its occurrences, ignoring the appropriate word meaning in various contexts, the *contextualized embeddings* are computed for each word occurrence, taking into account the whole sentence. Three ways of computing such contextualized embeddings have been proposed: ELMo [27], BERT [5] and Flair [1], along with precomputed models.

Peters et al. (2018) [27] obtain the proposed embeddings, called *ELMo*, from internal states of deep bidirectional language model, pretrained on a large corpus. Akbik et al. (2018) [1] introduced *Flair*, contextualized word embeddings obtained from internal states of a character-level bidirectional language model, thus significantly increasing state of the art of POS tagging, chunking and NER tasks. Last, but not least, Devlin et al. (2018) [5] employ a Transformer [37] to compute contextualized embeddings from preceeding and following context at the same time, at the cost of increased processing costs. The new *BERT* embeddings achieved state-of-the-art results in eleven natural language tasks.

Using two of these methods, for which precomputed models for Czech are available, namely BERT and Flair, we present our models for four NLP tasks: part-of-speech (POS) tagging, lemmatization, dependency parsing and named entity recognition (NER). Adding the contextualized embeddings as optional inputs in strong artificial neural network baselines, we report state-of-the-art results in these four tasks.

2 Related Work

As for the Prague Dependency Treebank (PDT) [13], most of the previous works are non-neural systems with one exception of [19] who hold the state of the art for Czech POS tagging and lemmatization, achieved with the recurrent neural network (RNN) using end-to-end trainable word embeddings and character-level word embeddings. Otherwise, Spoustová et al. (2009) [31] used an averaged perceptron for POS tagging. For parsing the PDT, Holan and Zabokrtský (2006) [16] and Novák and Žabokrtský (2007) [26] used a combination of non-neural parsing techniques .

In the multilingual shared task *CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* [39], raw text is processed and the POS tagging, lemmatization and dependency parsing are evaluated on the Universal Dependencies (UD) [24]. Czech is one of the 57 evaluated languages. Interestingly, all 26 participant systems employed the artificial neural networks in some way. Of these, 3 participant systems used (a slightly modified variant of) the only newly presented contextualized embeddings called ELMo [27], most notably one of the shared task winners [3]. BERT and Flair were not available at the time.

For the Czech NER, Straková et al. (2016) [36] use an artificial neural network with word- and character-level word embeddings to perform NER on the Czech Named Entity Corpus (CNEC) [28,29,30].

3 Datasets

3.1 Prague Dependency Treebank 3.5

The *Prague Dependency Treebank 3.5* [13] is a 2018 edition of the core *Prague Dependency Treebank*. The Prague Dependency Treebank 3.5 contains the same texts as the previous versions since 2.0, and is divided into `train`, `dtest`, and `etest` subparts, where `dtest` is used as a development set and `etest` as a test set. The dataset consists of several layers – the morphological `m`-layer is the largest and contains morphological annotations (POS tags and lemmas), the analytical `a`-layer contains labeled dependency trees, and the `t`-layer is the smallest and contains tectogrammatical trees. The statistics of PDT 3.5 sizes is presented in Table 1.

A detailed description of the morphological system can be found in [11], a specification of the syntactic annotations has been presented in [10]. We note that in PDT, lemmas with the same word form are disambiguated using a number suffix – for example, English lemmas for the word forms `can` (noun) and `can` (verb) would be annotated as `can-1` and `can-2`.

In evaluation, we compute:

- POS tagging accuracy,
- lemmatization accuracy,
- unlabeled attachment score (UAS),
- labeled attachment score (LAS).

Part	Morphological m-layer		Analytical a-layer	
	Words	Sentences	Words	Sentences
Train	1 535 826	90 828	1 171 190	68 495
Development	201 651	11 880	158 962	9 270
Test	219 765	13 136	173 586	10 148

Table 1. Size of morphological and analytical annotations of PDT 3.5 train/development/test sets.

3.2 Universal Dependencies

The *Universal Dependencies* project [24] seeks to develop cross-linguistically consistent treebank annotation of morphology and syntax for many languages. We evaluate the Czech PDT treebank of UD 2.3 [25], which is an automated conversion of PDT 3.5 a-layer to Universal Dependencies annotation. The original POS tags are used to generate **UPOS** (universal POS tags), **XPOS** (language-specific POS tags, in this case the original PDT tags), and **Feats** (universal morphological features). The UD lemmas are the raw textual lemmas, so the discriminative numeric suffix of PDT is dropped. The dependency trees are converted according to the UD guidelines, adapting both the unlabeled trees and the dependency labels.

To compute the evaluation scores, we use the official *CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* [39] evaluation script, which produces the following metrics:

- **UPOS** – universal POS tags accuracy,
- **XPOS** – language-specific POS tags accuracy,
- **UFeats** – universal subset of morphological features accuracy,
- **Lemmas** – lemmatization accuracy,
- **UAS** – unlabeled attachment score, **LAS** – labeled attachment score,
- **MLAS** – morphology-aware LAS, **BLEX** – bi-lexical dependency score.

3.3 Czech Named Entity Corpus

The *Czech Named Entity Corpus 1.1* [28,29] is a corpus of 5 868 Czech sentences with manually annotated 33 662 Czech named entities, classified according to a two-level hierarchy of 62 named entities.

The *Czech Named Entity Corpus 2.0* [30] contains 8 993 Czech sentences with manually annotated 35 220 Czech named entities, classified according to a two-level hierarchy of 46 named entities.

We evaluate the NER task with the official CNEC evaluation script. Similarly to previous literature [28,36] etc., the script only evaluates the first round annotation classes for the CNEC 1.1. For the CNEC 2.0, the script evaluates all annotated classes.

4 Neural Architectures

All our neural architectures are recurrent neural networks (RNNs). The POS tagging, lemmatization and dependency parsing is performed with the *UDPipe 2.0* (Section 4.1) and NER is performed with our new sequence-to-sequence model (Section 4.2).

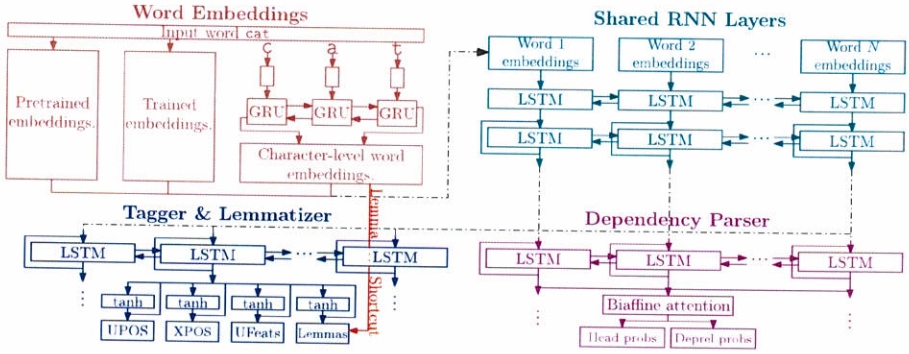

Fig. 1. UDPipe 2.0 architecture overview.

4.1 POS Tagging, Lemmatization, and Dependency Parsing

We perform POS tagging, lemmatization and dependency parsing using *UDPipe 2.0* [32], one of the three winning systems of the *CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* [39] and an overall winner of *The 2018 Shared Task on Extrinsic Parser Evaluation* [7]. An overview of this architecture is presented in Figure 1 and the full details of the architecture and the training procedure are available in [32].

POS Tagging and Lemmatization The tagger employs a standard bi-LSTM architecture. After embedding input words, three bidirectional LSTM [15] layers are performed, followed by a softmax output layers for POS tags and lemmas. While a classification output layer is natural for POS tags, we also apply it to lemmatization and generate lemmas by classifying the input words into lemma generation rules, therefore considering lemmatization as another tagging task.

We construct a lemma generation rule from a given form and lemma as follows:

- We start by finding the longest continuous substring of the form and the lemma. If it is empty, we use the lemma itself as the class.
- If there is a common substring of the form and the lemma, we compute the shortest edit script converting the prefix of the form into the prefix of the lemma, and the shortest edit script converting the suffix of the form to the suffix of the lemma. The edit scripts permit the operations `delete_current_char` and `insert_char(c)`.
- All above operations are performed case insensitively. To indicate correct casing of the lemma, we consider the lemma to be a concatenation of segments, where each segment is composed of either a sequence of lowercase characters, or a sequence of uppercase characters. We represent the lemma casing by encoding the beginning of every such segment, where the offsets in the first half of the lemma are computed relatively to the start of the lemma, and the offsets in the second half of the lemma are computed relatively to the end of the lemma.

Dependency Parsing The dependency parsing is again predicted using *UDPipe 2.0* architecture. After embedding input words, three bidirectional LSTM [15] layers are again performed, followed by a biaffine attention layer [6] producing labeled dependency trees.

In our evaluation we do not utilize gold POS tags and lemmas on the test set for dependency parsing. Instead, we consider three ways of employing them during parsing:

- not using them at all;
- adding predicted POS tags and lemmas on input;
- perform joint training of POS tags, lemmatization, and dependency parsing. In this case, we share first two bidirectional LSTM layers between the tagger and the parser.

Input Embeddings In our **baseline** model, we use the end-to-end word embeddings and also character-level word embeddings (bidirectional GRUs, [4,9,22] of dimension 256) trained specifically for the task.

Our architecture can optionally employ the following additional inputs

- **pretrained word embeddings (WE):** For the PDT experiments, we generate the word embeddings with `word2vec`[1] on a concatenation of large raw Czech corpora[2] available from the LINDAT/CLARIN repository.[3] For UD Czech, we use FastText word embeddings [2] of dimension 300, which we pretrain on Czech Wikipedia using segmentation and tokenization trained from the UD data.[4]
- **BERT** [5]: Pretrained contextual word embeddings of dimension 768 from the `Base` model.[5] We average the last four layers of the BERT model to produce the embeddings. Because BERT utilizes word pieces, we decompose UD words into appropriate subwords and then average the generated embeddings over subwords belonging to the same word.
- **Flair** [1]: Pretrained contextual word embeddings of dimension 4096.

POS Tags and Lemmas Decoding Optionally, we employ a morphological dictionary MorfFlex [12] during decoding. If the morphological dictionary is used, it may produce analyses for an input word as *(POS tag, lemma)* pairs. If any are generated, we choose the pair with maximum likelihood given by both the POS tag and lemmatization model.

4.2 Named Entity Recognition

We use a novel approach for nested named entity recognition (NER) to capture the nested entities in the Czech Named Entity Corpus.[6] The nested entities are encoded in

[1] With options `-size 300 -window 5 -negative 5 -iter 1 -cbow 0`.

[2] The concatenated corpus has approximately 4G words, two thirds of them from SYN v3 [14].

[3] `https://lindat.cz`

[4] We use `-minCount 5 -epoch 10 -neg 10` options to generate the embeddings.

[5] We use the BERT-Base Multilingual Uncased model from `https://github.com/google-research/bert`.

[6] Under review for ACL 2019.

a sequence and the problem of nested NER is then viewed as a sequence-to-sequence (seq2seq) problem, in which the input sequence consists of the input tokens (forms) and the output sequence of the linearized entity labels.

The system is a encoder-decoder architecture. The encoder is a bi-directional LSTM and the decoder is a LSTM. The encoded labels are predicted one by one by the decoder, until the decoder outputs the "`<eow>`" (end of word) label and moves to the next token. We use a hard attention on the word whose label(s) is being predicted.

We train the network using the lazy variant of the Adam optimizer [18], which only updates accumulators for variables that appear in the current batch,[7] with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We use mini-batches of size 8. As a regularization, we apply dropout with rate 0.5 and the word dropout replaces 20% of words by the unknown token to force the network to rely more on context. We did not perform any complex hyperparameter search.

In this model, we use the following word- and character-level word embeddings:

- **pretrained word embeddings:** We use the FastText [2] word embeddings of dimension 300 from the publicly available Czech model.[8]
- **end-to-end word embeddings:** We embed the input forms and lemmas (256 dimensions) and POS tags (one-hot).[9]
- **end-to-end character-level word embeddings:** We use bidirectional GRUs [4,9] of dimension 128 in line with [22]: we represent every Unicode character with a vector of dimension 128, and concatenate GRU outputs for forward and reversed word characters.

Optionally, we add the **BERT** [5] and the **Flair** [1] contextualized embeddings in the same way as in the UDPipe 2.0 (Section 4.1).

5 Results

5.1 POS Tagging and Lemmatization on PDT 3.5

The POS tagging and lemmatization results are presented in Table 2. The word2vec word embeddings (WE) considerably increase performance compared to the baseline, especially in POS tagging. When only Flair embeddings are added to the baseline, we also observe an improvement, but not as high. We hypothesise that the lower performance (in contrast with the results reported in [1]) is caused by the size of the training data, because we train the word2vec WE on considerably larger dataset than the Czech Flair model. However, when WE and Flair embeddings are combined, performance moderately increases, demonstrating that the two embedding methods produce at least partially complementary representations.

The BERT embeddings alone bring highest improvement in performance. Furthermore, combination with WE or Flair again yields performance increase. The best results

[7] `tf.contrib.opt.LazyAdamOptimizer` from `www.tensorflow.org`

[8] `https://fasttext.cc/docs/en/crawl-vectors.html`

[9] POS tagging and lemmatization done with MorphoDiTa [34], `http://ufal.mff.cuni.cz/morphodita`.

WE	BERT	Flair	Without Dictionary			With Dictionary		
			POS Tags	Lemmas	Both	POS Tags	Lemmas	Both
			96.88%	98.35%	96.21%	97.31%	98.80%	96.89%
✓			97.43%	98.55%	96.77%	97.59%	98.82%	97.18%
		✓	97.24%	98.49%	96.61%	97.54%	98.86%	97.14%
✓		✓	97.53%	98.63%	96.91%	97.69%	98.88%	97.28%
	✓		97.67%	98.63%	97.02%	97.91%	98.94%	97.51%
✓	✓		97.86%	98.69%	97.21%	98.00%	98.96%	97.59%
	✓	✓	97.80%	98.67%	97.16%	98.00%	98.96%	97.59%
✓	✓	✓	**97.94%**	**98.75%**	**97.31%**	**98.05%**	**98.98%**	**97.65%**
Morče (2009) [31]			—	—	—	*95.67%*†	—	—
MorphoDiTa (2016) [35]			—	—	—	*95.55%*	*97.85%*	*95.06%*
LemmaTag (2018) [19]			*96.90%*	*98.37%*	—	—	—	—

Table 2. POS tagging and lemmatization results (accuracy) on PDT 3.5. **Bold** indicates the best result, *italics* related work. †Reported on PDT 2.0, which has the same underlying corpus, with minor changes in morphological annotation (our model results differ at 0.1% on PDT 2.0).

POS Tags, Lemmas	BERT	Flair	UAS (unlabeled attachment score)	LAS (labeled attachment score)	POS Tags	Lemmas
			91.16%	87.35%	—	—
		✓	91.38%	87.69%	—	—
	✓		92.75%	89.46%	—	—
	✓	✓	92.76%	89.47%	—	—
Predicted on input	✓	✓	92.84%	89.62%	—	—
Joint prediction			91.69%	88.16%	*97.33%*	*98.42%*
Joint prediction		✓	91.89%	88.42%	*97.48%*	*98.42%*
Joint prediction	✓		93.01%	89.74%	*97.62%*	*98.49%*
Joint prediction	✓	✓	**93.07%**	**89.89%**	*97.72%*	*98.51%*
Gold on input	✓	✓	*92.95%*	*89.89%*	—	—
POS tagger trained on 3.5 a-layer			—	—	*97.82%*	*98.66%*

Table 3. Dependency tree parsing results on PDT 3.5 a-layer. **Bold** indicates the best result, *italics* POS tagging and lemmatization results. For comparison, we report results of a parser trained using gold POS tags and lemmas, and of a tagger trained on a-layer (both also in *italics*).

System	UAS (unlabeled attachment score)	LAS (labeled attachment score)
Our best system (joint prediction, BERT, Flair)	**93.10%**	**89.93%**
Holan and Žabokrtský (2006) [16]	*85.84%*	—
Novák and Žabokrtský (2007) [26]	*84.69%*	—
Koo et al. (2010) [21]†	*87.32%*	—
Treex framework (using MST parser&manual rules) [38]‡	*83.93%*	*77.04%*
PDT 2.0 subset in CoNLL 2007 shared task; manually annotated POS tags available.		
Nakagawa (2007) [23]	*86.28%*	*80.19%*
PDT 2.0 subset in CoNLL 2009 shared task; manually annotated POS tags available.		
Gesmundo et al. (2009) [8]	—	*80.38%*

Table 4. Dependency tree parsing results on PDT 2.0 a-layer. **Bold** indicates the best result, *italics* related work. †Possibly using gold POS tags. ‡Results as of 23 Mar 2019.

are achieved by exploiting all three embedding methods, substantially exceeding state-of-the-art results.

Utilization of morphological dictionary improves prediction accuracy. However, as the performance of a model itself increases, the gains obtained by the morphological dictionary diminishes – for a model without any pretrained embeddings, morphological dictionary improves POS tagging by and lemmatization by 0.43% and 0.45%, while the best performing model gains only 0.11% and 0.23%.

5.2 Dependency Parsing on PDT 3.5

The evaluation of the contextualized embeddings methods as well as various ways of POS tag utilization is presented in Table 3. Without POS tags and lemmas, the Flair embeddings bring only a slight improvement in dependency parsing when added to WE. In contrast, BERT embeddings employment results in substantial gains, increasing UAS and LAS by 1.6% and 2.1%. A combination of BERT and Flair embeddings does not result in any performance improvement, demonstrating that BERT syntactic representations encompass the Flair embeddings.

When introducing POS tags and lemmas predicted by the best model from Section 5.1 as inputs for dependency parsing, the performance increases only slightly. A better way of POS tags and lemmas exploitation is achieved in a joint model, which predicts POS tags, lemmas, and dependency trees simultaneously. Again, BERT embeddings bring significant improvements, but in contrast to syntax parsing only, adding Flair embeddings to BERT results in moderate gain – we hypothesise that the increase is due to the complementary morphological information present in Flair embeddings (cf. Section 5.1). Note that the joint model achieves better parsing accuracy than the one given gold POS tags and lemmas on input. However, the POS tags and lemmas predicted by the joint model are of slightly lower quality compared to a standalone tagger of the best configuration from Section 5.1.

Table 4 compares our best model with state-of-the-art results on PDT 2.0 (note that some of the related work used only a subset of PDT 2.0 and/or utilized gold morphological annotation). To our best knowledge, research on PDT parsing was performed mostly in the first decade of this century, therefore even our baseline model substantially surpasses previous works. Our best model with contextualized embeddings achieves nearly 50% error reduction both in UAS and LAS.

5.3 POS Tagging, Lemmatization and Dependency Parsing on Universal Dependencies

Table 5 shows the performance of analyzed embedding methods in a joint model performing POS tagging, lemmatization, and dependency parsing on Czech PDT UD 2.3 treebank. This treebank is derived from PDT 3.5 `a`-layer, with original POS tags kept in XPOS, and the dependency trees and lemmas modified according to UD guidelines.

We observe that the word2vec WEs perform similarly to Flair embeddings in this setting. Our hypothesis is that the word2vec WEs performance loss (compared to WEs in Section 5.1) is caused by using a considerably smaller raw corpus to pretrain the WEs (Czech Wikipedia with 785M words, compared to 4G words used in Section 5.1), due

WE	BERT	Flair	UPOS	XPOS	UFeats	Lemmas	UAS	LAS	MLAS	BLEX
			99.06	96.73	96.69	98.80	92.93	90.75	84.99	87.68
✓			99.18	97.28	97.23	99.02	93.33	91.31	86.15	88.60
		✓	99.16	97.17	97.13	98.93	93.33	91.33	86.19	88.56
✓		✓	99.22	97.41	97.36	99.07	93.48	91.49	86.62	88.89
	✓		99.25	97.46	97.41	99.00	94.26	92.34	87.53	89.79
✓	✓		99.31	97.61	97.55	99.06	94.27	92.34	87.75	89.91
✓	✓	✓	**99.34**	**97.71**	**97.67**	**99.12**	**94.43**	**92.56**	**88.09**	**90.22**
CoNLL 2018 Shared Task results on Czech PDT UD 2.2 treebank, without gold segmentation and tokenization.										
Our best system			**99.23**	**97.49**	**97.43**	**99.01**	**93.57**	91.64	**87.15**	**89.31**
HIT-SCIR (2018) [3]			*99.05*	*96.92*	*92.40*	*97.78*	*93.44*	***91.68***	*80.57*	*87.91*
TurkuNLP (2018) [17]			*98.74*	*95.44*	*95.22*	*98.50*	*92.57*	*90.57*	*83.16*	*87.63*

Table 5. Czech PDT UD 2.3 results for POS tagging (UPOS: universal POS, XPOS: language-specific POS, UFeats: universal morphological features), lemmatization and dependency parsing (UAS, LAS, MLAS, and BLEX scores). **Bold** indicates the best result, *italics* related work.

BERT	Flair	CNEC 1.1		CNEC 2.0	
		Types	Supertypes	Types	Supertypes
		82.96	86.80	80.47	85.15
	✓	83.55	87.62	81.65	85.96
✓		86.73	89.85	**86.23**	**89.37**
✓	✓	**86.88**	**89.91**	85.52	89.01
Konkol et al. (2013) [20]		–	*79.00*	–	–
Straková et al. (2013) [33]		*79.23*	*82.82*	–	–
Straková et al. (2016) [36]		*81.20*	*84.68*	*79.23*	*82.78*

Table 6. Named entity recognition results (F1) on the Czech Named Entity Corpus. **Bold** indicates the best result, *italics* related work.

to licensing reasons. BERT embeddings once more deliver the highest improvement, especially in dependency parsing, and our best model employs all three embedding methods.

In the previous ablation experiments, we used the gold segmentation and tokenization in the Czech PDT UD 2.3 treebank. For comparison with state of the art, Czech PDT UD 2.2 treebank without gold segmentation and tokenization is used in evaluation, according to the CoNLL 2018 shared task training and evaluation protocol. Our system surpasses previous works substantially in all metrics except UAS and LAS, where it achieved results comparable to HIT-SCIR system [3] (which nevertheless employs ensembling improving results by 0.2% according to [3]).

Comparing the results with a joint tagging and parsing PDT 3.5 model from Table 1, we observe that the XPOS results are nearly identical as expected. Lemmatization on the UD treebank is performed without the discriminative numeric suffixes (see Section 3.2) and therefore reaches better performance. Both UAS and LAS are also better on the UD treebank, which we assume is caused by the different annotation scheme.

5.4 Named Entity Recognition

Table 6 shows NER results (F1 score) on CNEC 1.1 and CNEC 2.0. Our sequence-to-sequence (seq2seq) model which captures the nested entities, clearly surpasses the current Czech NER state of the art. Furthermore, significant improvement is gained when adding the contextualized word embeddings (BERT and Flair) as optional input to the LSTM encoder. The strongest model is a combination of the sequence-to-sequence architecture with both BERT and Flair contextual word embeddings.

6 Conclusion

We have presented an evaluation of two contextualized embeddings methods, namely BERT and Flair. By utilizing these embeddings as input to deep neural networks, we have achieved state-of-the-art results in several Czech text processing tasks, namely in POS tagging, lemmatization, dependency parsing and named entity recognition.

7 Acknowledgements

The work described herein has been supported by the VIADAT project, project No. DG16P02R019, by the Czech Ministry of Culture; data has been provided by the LINDAT/CLARIN repository, which is supported by infrastructural projects of the Czech MSMT No. CZ.02.1.01/0.0/0.0/16_013/0001781 (OP VVV) and LM2015071 (VI).

References

1. Akbik, A., Blythe, D., Vollgraf, R.: Contextual String Embeddings for Sequence Labeling. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 1638–1649. Association for Computational Linguistics (2018)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics 5, 135–146 (2017)
3. Che, W., Liu, Y., Wang, Y., Zheng, B., Liu, T.: Towards better ud parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 55–64. Association for Computational Linguistics (2018)
4. Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y.: On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. CoRR (2014)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRR abs/1810.04805 (2018)
6. Dozat, T., Manning, C.D.: Deep Biaffine Attention for Neural Dependency Parsing. CoRR abs/1611.01734 (2016)
7. Fares, M., Oepen, S., Øvrelid, L., Björne, J., Johansson, R.: The 2018 Shared Task on Extrinsic Parser Evaluation: On the Downstream Utility of English Universal Dependency Parsers. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 22–33. Association for Computational Linguistics (2018)

8. Gesmundo, A., Henderson, J., Merlo, P., Titov, I.: A latent variable model of synchronous syntactic-semantic parsing for multiple languages. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task. pp. 37–42. Association for Computational Linguistics, Boulder, Colorado (Jun 2009)
9. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks pp. 5–6 (2005)
10. Hajič, J.: Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In: Hajičová, E. (ed.) Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová, pp. 106–132. Karolinum, Charles University Press, Prague, Czech Republic (1998)
11. Hajič, J.: Disambiguation of Rich Inflection: Computational Morphology of Czech. Karolinum Press (2004)
12. Hajič, J., Hlaváčová, J.: MorfFlex CZ 161115 (2016), http://hdl.handle.net/11234/1-1834, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), aculty of Mathematics and Physics, Charles University
13. Hajič, J., et al.: Prague dependency treebank 3.5 (2018), http://hdl.handle.net/11234/1-2621, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University
14. Hnátková, M., Křen, M., Procházka, P., Skoumalová, H.: The syn-series corpora of written czech. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC'14). pp. 160–164. European Language Resources Association (ELRA), Reykjavik, Iceland (May 2014)
15. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Comput. 9(8), 1735–1780 (November 1997)
16. Holan, T., Žabokrtský, Z.: Combining czech dependency parsers. In: Sojka, P., Kopeček, I., Pala, K. (eds.) Text, Speech and Dialogue. pp. 95–102. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
17. 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: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 133–142. Association for Computational Linguistics, Brussels, Belgium (October 2018)
18. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimization. International Conference on Learning Representations (12 2014)
19. Kondratyuk, D., Gavenčiak, T., Straka, M., Hajič, J.: Lemmatag: Jointly tagging and lemmatizing for morphologically rich languages with brnns. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4921–4928. Association for Computational Linguistics (2018)
20. Konkol, M., Konopík, M.: CRF-based Czech named entity recognizer and consolidation of Czech NER research. In: Text, Speech, and Dialogue. pp. 153–160. Springer Berlin Heidelberg (2013)
21. Koo, T., Rush, A.M., Collins, M., Jaakkola, T., Sontag, D.: Dual decomposition for parsing with non-projective head automata. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. pp. 1288–1298. Association for Computational Linguistics, Cambridge, MA (October 2010)
22. Ling, W., Luís, T., Marujo, L., Astudillo, R.F., Amir, S., Dyer, C., Black, A.W., Trancoso, I.: Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. CoRR (2015)
23. Nakagawa, T.: Multilingual dependency parsing using global features. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007. pp. 952–956. Association for Computational Linguistics, Prague, Czech Republic (Jun 2007)

24. Nivre, J., de Marneffe, M.C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D.: Universal Dependencies v1: A multilingual treebank collection. In: Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016). pp. 1659–1666. European Language Resources Association, Portorož, Slovenia (2016)
25. Nivre, J., et al.: Universal dependencies 2.3 (2018), http://hdl.handle.net/11234/1-2895, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University
26. Novák, V., Žabokrtský, Z.: Feature engineering in maximum spanning tree dependency parser. In: Matoušek, V., Mautner, P. (eds.) Lecture Notes in Artificial Intelligence, Proceedings of the 10th International Conference on Text, Speech and Dialogue. Lecture Notes in Computer Science, vol. 4629, pp. 92–98. Springer, Berlin / Heidelberg (2007)
27. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep Contextualized Word Representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 2227–2237. Association for Computational Linguistics (2018)
28. Ševčíková, M., Žabokrtský, Z., Krůza, O.: Named Entities in Czech: Annotating Data and Developing NE Tagger. In: Matoušek, V., Mautner, P. (eds.) Lecture Notes in Artificial Intelligence, Proceedings of the 10th International Conference on Text, Speech and Dialogue. Lecture Notes in Computer Science, vol. 4629, pp. 188–195. Springer, Berlin / Heidelberg (2007)
29. Ševčíková, M., Žabokrtský, Z., Straková, J., Straka, M.: Czech named entity corpus 1.1 (2014), http://hdl.handle.net/11858/00-097C-0000-0023-1B04-C, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University
30. Ševčíková, M., Žabokrtský, Z., Straková, J., Straka, M.: Czech named entity corpus 2.0 (2014), http://hdl.handle.net/11858/00-097C-0000-0023-1B22-8, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University
31. Spoustová, D.j., Hajič, J., Raab, J., Spousta, M.: Semi-Supervised Training for the Averaged Perceptron POS Tagger. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009). pp. 763–771. Association for Computational Linguistics (Mar 2009)
32. Straka, M.: UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In: Proceedings of CoNLL 2018: The SIGNLL Conference on Computational Natural Language Learning. pp. 197–207. Association for Computational Linguistics, Stroudsburg, PA, USA (2018)
33. Straková, J., Straka, M., Hajič, J.: A New State-of-The-Art Czech Named Entity Recognizer. In: Text, Speech, and Dialogue, Lecture Notes in Computer Science, vol. 8082, pp. 68–75. Springer Berlin Heidelberg (2013)
34. Straková, J., Straka, M., Hajič, J.: Open-source tools for morphology, lemmatization, POS tagging and named entity recognition. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 13–18. Johns Hopkins University, Baltimore, MD, USA, Association for Computational Linguistics, Stroudsburg, PA, USA (2014)
35. Straková, J., Straka, M., Hajič, J.: Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 13–18. Johns Hopkins University, Baltimore, MD, USA, Association for Computational Linguistics (2014)
36. Straková, J., Straka, M., Hajič, J.: Neural Networks for Featureless Named Entity Recognition in Czech. In: Text, Speech, and Dialogue: 19th International Conference, TSD 2016,

Brno , Czech Republic, September 12-16, 2016, Proceedings. pp. 173–181. Springer International Publishing (2016)
37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. CoRR abs/1706.03762 (2017)
38. Žabokrtský, Z.: Treex – an open-source framework for natural language processing. In: Lopatková, M. (ed.) Information Technologies – Applications and Theory. vol. 788, pp. 7–14. Univerzita Pavla Jozefa Šafárika v Košiciach, Univerzita Pavla Jozefa Šafárika v Košiciach, Košice, Slovakia (2011)
39. Zeman, D., Ginter, F., Hajič, J., Nivre, J., Popel, M., Straka, M.: CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Association for Computational Linguistics, Brussels, Belgium (2018)