Neural Tagger for Czech Language: Capturing Linguistic Phenomena in Web Corpora

Zuzana Nevěřilová and Marie Stará

Natural Language Processing Centre
Faculty of Informatics
Botanická 68a, Brno, Czech Republic

Abstract. We propose a new tagger for the Czech language and particularly for the tagset used for annotation of corpora of the TenTen family. The tagger is based on neural networks with pretrained word embeddings. We selected the newest Czech Web corpus of the TenTen family as training data, but we removed sentences with phenomena that were often annotated incorrectly. We let the tagger to learn the annotation of these phenomena on its own. We also experimented with the recognition of multi-word expressions since this information can support the correct tagging.
We evaluated the tagger on 6,950 sentences (84,023 tokens) from the `cstenten17` corpus and achieved 75.25% accuracy when compared by tags. When compared by attributes, we achieved 91.62% accuracy; the accuracy of POS tag prediction is 96.5%.

Keywords: Czech Tagger, Multi-word Expressions, Pretrained Word Embeddings

1 Introduction

Currently, the processing pipeline for Czech Web corpora of the TenTen family [2] uses the tagger `desamb` [7], based on the morphological analyser `majka` [11], the guesser, and inductive logical programming. This approach has been sufficient for many texts that follow the grammatical, syntactic, and orthographic rules of the language. With the rise of Web corpora, a more flexible solution is desired. The basic requirements comprise:

1. adaptability to typos/incorrect orthography
2. adaptability to neologisms
3. ability to distinguish foreign language injections
4. ability to annotate foreign proper nouns
5. adaptability to newly appearing syntactic patterns
6. ability to recognize multi-word expressions (MWEs)

While the requirements 1, 2, and 4 are to a large extent fulfilled by the current pipeline, requirement 3 has been solved in several works, and requirements 5 and 6 are currently not solved for the Czech language. We propose a new tagger

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2019, pp. 23–32, 2019.

based on neural networks that is able to learn from the current Web corpus of Czech and thus is able to reflect the change in the language use.

This work aims to be one of the steps towards an adaptable MWE-aware tagger for Czech.

1.1 Paper Outline

Section 2 describes current taggers for Czech language, Section 3 describes the training examples selection process. In Section 4, we provide detailed information on the neural network architecture and parameters. Section 5 provides evaluation results and error analysis. In Section 6, we plan further work in the area.

2 Related Work

Historically, for Czech, two different tagsets are used, unfortunately not easily convertible. The tagset used for annotation of corpora of the TenTen family is an attributional system used in the tagger `desamb` [7] and the morphological analyser `majka` [11]. A detailed description of the tagset is in [3].

For the positional tagset, the most widely used tagger MorphoDiTa was developed [8]. The authors report 95.75% accuracy in POS tagging. MorphoDiTa replaces its predecessor Morče based on the same algorithm.

The focus of another tagger, MUMULS [10], is in verbal MWE identification. The tagger annotates POS and MWEs. The authors report that token F1 is 73% on Czech language, which is only one among 10 languages the tagger is able to process.

3 Training Data

As training data for our tagger, we selected first million sentences from the Web corpus `cstenten17` [9]. In the version `mj2` we used, the corpus was tagged using `majka+desamb` pipeline `v2`[1].

From previous versions of the `cstenten` (formerly `cztenten`) corpus, some problematic phenomena were known. In our previous work [5], we identified the incorrect annotation of inter-lingual homographs (words that exist in different languages but have a different function). For example, the word *top* is an English noun or adjective, while in Czech, it is an imperative form of *to drawn* or *to stoke up*.

The second group of problems is caused by the fact that the guesser is intended to be used for Czech out-of-vocabulary words (OOVs). Its performance is much lower in the case of foreign names (person names, brand names, place names, etc.). A nice example of such annotation is the word *Wikipedia* often annotated as plural genitive (probably because of its Latin-like ending).

[1] Further information in `https://www.sketchengine.eu/cstenten-czech-corpus/`

The third group of problematic annotation results in guessed lemmata for some groups of Czech OOVs that never appear as words in the corpus. There can be a case that a word is never used in its base form, but it is very likely that these lemmata were simply guessed incorrectly. Examples of such annotations are words such as *šedat* or *mývalit*, incorrectly recognized as verbs instead of adjectives *šedý*, *mývalí*.

Because of the systematic nature of the incorrect annotations of the phenomena mentioned above, we filtered out sentences containing these phenomena. We created a blacklist composed of OOV proper names, interlingual homographs, and guessed lemmata that never appear in the corpus as words. We kept 606,351 sentences from which we selected a random sample.

4 Training the Model

Tagging is a sequence processing task, therefore a usual neural tagger is composed of the input layer, embedding layer, one or several hidden recurrent layers, and the output layer. The usual scheme is depicted in Figure 1. The input sequence I represents the maximum number of tokens s, embedding layers encodes the tokens into vectors, the output layer O represents tags for tokens, its length c is the number of possible tags.

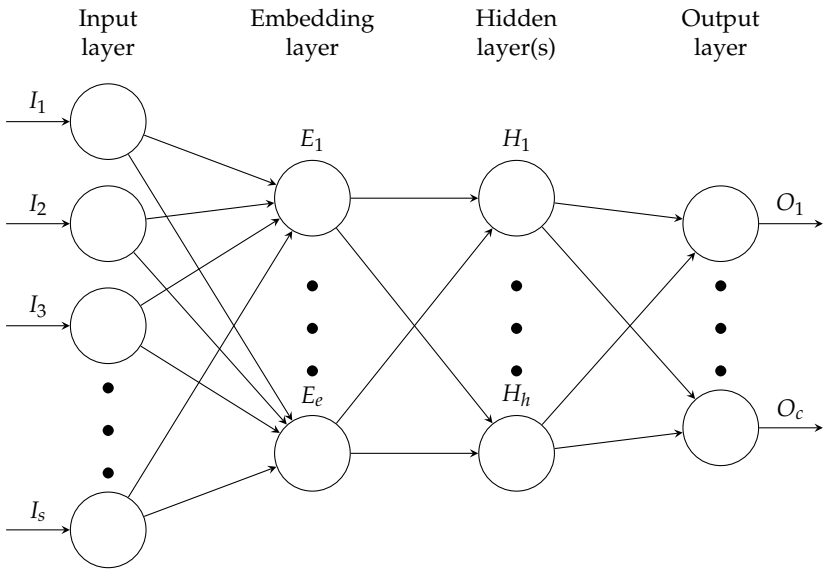

Fig. 1: Usual architecture for a tagger. The input sequence has length s (the number of tokens), the output layer length c is the number of possible tags.

4.1 Tagset Encoding

Since Czech taggers have to annotate many grammatical categories (that have many possible values), the number of possible tags is too large. In the Universal dependencies, the `ajka` tagset contains 2,176 tags[2]. For comparison, the Penn Tree Bank tagset in Universal Dependencies has 48 different tags[3]. Our cleaned sample of 606k sentences uses 1,537 different tags. For experimental settings, we removed the following attributes:

- stylistic subclassification (attribute `w`)
- subclassification of pronouns (attribute `x`)
- subclassification of adverbs (attribute `y`)
- stylistic subclassification (attribute `w`)
- punctuation subclassification (attribute `x`)
- verb aspect (the attribute `a`)

After this tagset reduction, we split every tag into attributes and reformulate the problem. In typical cases, tagging is a classification problem with one-hot output vector. In our setting, each token belongs to n classes where $n = 0, ..., m$ and m is the maximum number of attributes in one tag. For example, a noun in nominative singular masculine inanimate (tag `k1gInSc1`) has four classes (noun, nominative, singular, masculine inanimate). Using this technique, we reduce the number of tags to 44. On the other hand, the implementation is slightly more complicated, since the output is not a sequence of one-hot vectors but a sequence of multi-hot vectors.

4.2 Pretrained Embeddings

Pretrained embeddings are very popular since they can be easily reused to capture the semantic relations between words. Using pretrained embeddings usually improves the tagger results significantly. The usual procedure is to set pretrained embeddings as input weights of the embedding layer E (see Figure 1) and set the layer non-trainable. The advantage is ease of implementation and the final model size, the disadvantage is the OOV problem, since the neural network contains only embeddings of the training examples.

Embedding models can be calculated in several ways, and pretrained models are available for many languages, mostly trained on Wikipedia data. We use `fasttext` [1] since it contains subword information which supports language modeling tasks significantly.

Inspired by [6], we incorporate the `fasttext` model directly into the input layer. We did not use the `Embedding` layer provided by TensorFlow Keras[4] because we implemented the embedding layer on our own. The consequence is that every input sequence of tokens has to be converted into a sequence of

[2] `https://universaldependencies.org/tagset-conversion/cs-ajka-uposf.html`
[3] `https://universaldependencies.org/tagset-conversion/en-penn-uposf.html`
[4] `https://www.tensorflow.org/guide/keras/overview`

embeddings using the same `fasttext` model. On the other hand, the advantage of this approach is a massive improvement of classification since we completely avoid the OOV problem.

4.3 Neural Network Architecture and Parameters

We set up a neural network with two bidirectional long short term memory network (Bi-LSTM) layers, both with spatial dropout. We limited the maximum sentence length to 20 tokens. For longer sentences, the prediction has to use a sliding window. The size of the two Bi-LSTM layers are 512 and 128, respectively, the size of the output layer is 44 (the number of possible attributes in tags). A scheme of the architecture is in Figure 2.

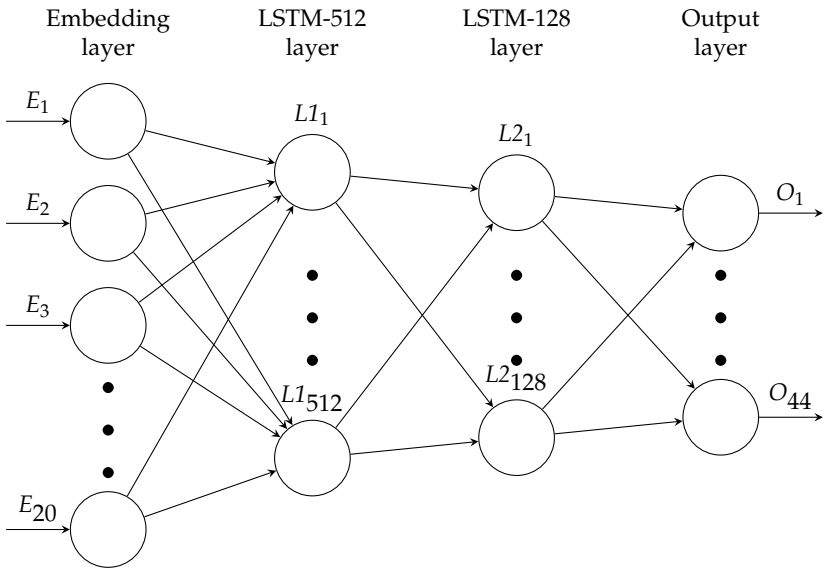

Fig. 2: Architecture of the tagger: the input layer already contains word embeddings, two Bi-LSTM layers follow with the dropout layers (not in the schema).

We used sigmoid activations for all layers and Adam optimizer together with binary cross-entropy loss. The input sequences were padded by zero vectors. We experimented with inserting tag distribution as input weights of the last layer, but those did not improve the network results.

The training took 9 epochs on 180,000 samples split into 171,000/19,000 train/test data. The measured accuracy ended up at 99.38%, and the measured validation loss went down to 1.70%. The results are so optimistic mainly because

of the zero padding since it is easy for the network to learn the tags for zero vectors.

The output layer simply performs rounding, so all values equal to or greater than 0.5 are converted to tag attributes. Furthermore, post-processing selects only one attribute in case several values of the same attribute have values over 0.5. For example, if the tagger predicts 0.56 for attribute `c1` and 0.65 for attribute `c4`, the post-processing selects `c4`. We do not force the tagger to predict all possible attribute values of the tag, so it can happen that the tagger predicts no attribute.

The model size is 18MB, but for prediction, the `fasttext` model (7GB) is needed in addition. According to the authors of `fasttext`, the model size can be shrunk using quantization. We did not solve the issue yet.

5 Evaluation

We performed a detailed quantitative evaluation of the tagger on 6,950 sentences not present in the training set but present in the cleaned data. The tag accuracy was 75.25% if measured by an exact match. Since the tagger is not forced to predict all possible attributes, we also measured the submatch accuracy, i.e., the case where the predicted tag is contained in the true tag. We achieved 87.62% submatch accuracy. Finally, we also measured the match between the attributes of the predicted and true tags. Here, we achieved 91.62% accuracy. Accuracy on the POS tag (the `k` tag) was 96,5%, the tagger did not predict any tag in 1,2% cases.

5.1 Quantitative Error Analysis

Not all tagging errors are equally serious. In the cases we mentioned in Section 3, the error severity is caused mainly by incorrect part-of-speech (POS) tagging since the POS tag is used in further statistical computations. Another type of errors, known in the corpus annotation, is incorrect gender and incorrect case (mainly distinction between nominative and accusative). These types of errors can also affect further processing, e.g., syntactic analysis.

In Figures 3 and 4, we provide confusion matrices calculated separately for attributes POS (`k`), gender (`g`), number (`n`), and case (`c`). The numbers represent hundreds of examples from the test data.

From confusion matrices, we can observe that misclassification of the POS tag does not occur very often. The most confused classes are nouns and interjections, nouns and adjectives, and adjectives and verbs. The noun-adjective uncertainty can be caused by nominalization (substantivization of adjectives) as well as by English borrowings with noun modifiers. The adjective-verb uncertainty originates in the similarity between adjectives and N-type past participles.

The second important observation concerns grammatical cases, where the nominative (case 1) and accusative (case 4) are often interchanged. This type of error is known already from the current annotation pipeline. Its solution is, therefore, very challenging since we know the training data is not clean enough.

From Figure 4, it can be seen number and especially gender are the most difficult attributes to annotate. Surprisingly, the dominant gender is feminine, and also the highest number of confusion is between feminine and the other genders. For verb attributes (grammatical mood and person) as well as for adjective and adverb attribute degree, the confusion matrices do not show much interesting information, so we do not provide them.

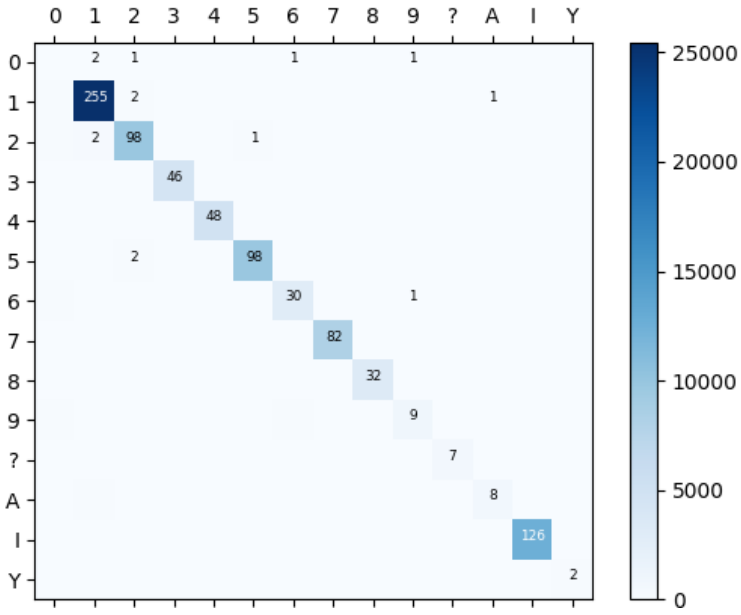

Fig. 3: Confusion matrix for the POS (attribute `k`).

5.2 Multi-word Expression Discovery

Using `fasttext`, we experimented with MWE discovery. We observed that in case of words a and b that are part of an MWE, their vectors are more similar to the vector of concatenated word $a + b$ than random words. We, therefore, calculate similarity matrices for each pair of neighboring words. If the sum of all elements in the similarity matrix is above a threshold t, we consider the pair to be an MWE. We only excluded punctuation from this calculation; however, we want to experiment with the grammatical information obtained by the tagger. Currently, we use the thresholds between $t = 7.1$ and $t = 7.3$. In the examples below, the identified MWEs are *Sulfid bismutitý* (meaning *bismuth(III) sulfide*), with threshold $t = 7.3$. With $t = 7.1$, the identified MWEs are also *jsou zveřejněny* (meaning *are published*), *and Cons*, *Hitch Hiking*, *The Wall*.

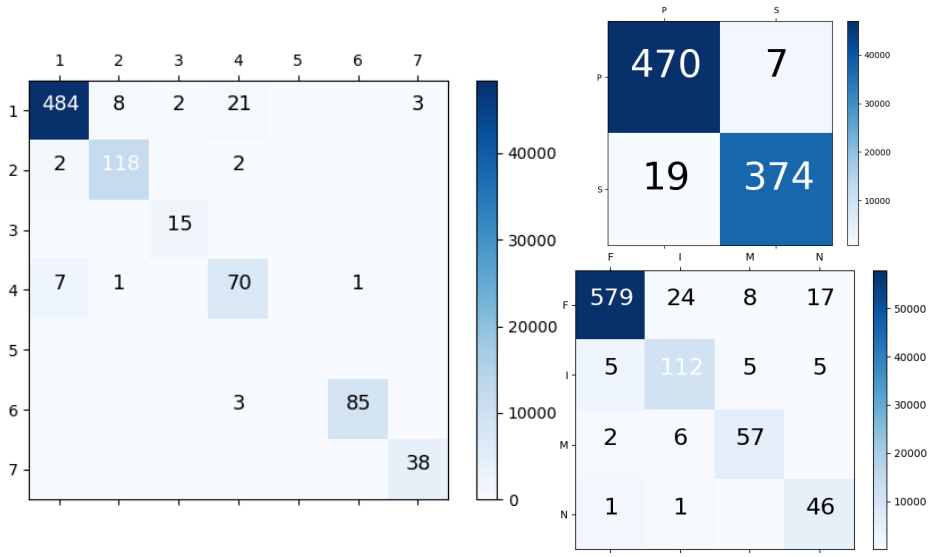

Fig. 4: Confusion matrices for the case, number, and gender (attributes `c`, `n`, and `g` respectively. N.B. that nouns, adjectives, pronouns, and numerals distinguish cases, number is an attribute also for verbs.

5.3 Manual Evaluation on Small Sample

The aim of the work is to outperform the current annotation pipeline, especially in the case of foreign words and inter-lingual homographs. To test this, we manually annotated a few random sentences from the corpus `bulky` [4]. The corpus is a subcorpus of the corpus `cztenten12` containing problematic phenomena such as foreign names and inter-lingual homographs. Figure 5 presents the outputs on five sentences. It can be seen that a serious error – incorrect POS tag – occurs 4× in case `desamb`, twice in case of our solution. Most other errors occur in the gender and case attributes. Some differences in the tagging are subject to discussion, for example, how to annotate foreign words or MWEs that play the role of a noun (e.g., the word *and* inside a multi-word named entity).

6 Conclusion and Future Work

We present a new tagger for Czech trained on the Web corpus `cztenten17` annotated using the `majka` tagset. We show that the neural approach is a promising direction, especially when combined with pretrained word embeddings containing subword information. The current POS tag accuracy is slightly above the results published for the tagger MorphoDiTa; however, the comparison was not made on the same data set.

	Do	vyhledávacího	pole	zadejte	Texas	HoldEm	Poker	.[5]
desamb	k7c2	k2gNnSc2d1	k1gNnSc2	k5mRp2nP	**kA**	k1gInS**c7**	k1gInSc1	kIx.
our	k7c2	k2gNnSc2*	k1gNnSc2	k5*p2*	k1gInSc1	k1gInSc1	k1gInSc1	kIx.

	Výsledky	jsou	zveřejněny	v	časopise	BMC	Biology	.[6]
desamb	k1gInPc1	k5mIp3nP	k5mNgInP	k7c6	k1gInSc6	kA	k1**gMnPc4**	kIx.
our	k1gInPc1	k5mIp3nP	k5mNgInP	k7c6	k1gInSc6	kA	k1nS	kIx.

	Lady	Murasaki	ztělesnila	čínská	herečka	Gong	Li[7]
desamb	k1gFnPc4	k1gFnPc4	k5mAgFnS	k2gFnSc1d1	k1gFnSc1	k1**gI**nS**c4**	**k8xS**
our	k1gFnP**c1**	k1gFnP**c1**	k5mAgFnS	k2gFnSc1d1	k1gFnSc1	k1gFnS	k1gFnS

	Album	The	Pros	and	Cons	of	Hitch	Hiking	…
desamb	k1gNnSc4	k1gFnS**c2**	k1gFnP**c2**	k?	k1gInS**c1**	k?	k1gInS**c1**	k1gInSc4	
our	k1gNnS**c1**	k1nS**c1**	k1nS**c1**	k1nS	k1nS**c1**	kA	k1gInS**c1**	k1gInS**c1**	
…	psal	Waters	v	roce	1978	současně	s	The	Wall[8]
desamb	k5mA**gI**nS	k1**gI**nSc1	k7c6	k1gInSc6	k4	k6d1	k7c7	k1gFnSc7	k1gFnP**c2**
our	k5mAgMnS	k1**gI**nSc1	k7c6	k1gInSc6	k4	k6d1	k7c7	k1nS	k1nS**c1**

	Sulfid	bismutitý	Bi2S3	je	tmavěhnědá	(	pokud	se …
desamb	k1gInSc1	k2g**Mn**Sc1d1	k1g**M**nSc1	k5mIp3nS	k2**gNnPc4**d1	kIxX	k8	k3c4
our	k1gInSc1	**k1gI**nSc1	k1nSc1	k5mIp3nS	k2*nSc1d1	kIxX	k8	k3c4

	připravuje	srážením	se	sirovodíkem	)	nebo	šedá	látka[9]
desamb	k5mIp3nS	k1gNnSc7	**k3c4**	k1gInSc7	kIxX	k8	**k5mIp3nS**	k1gFnSc1
our	k5mIp3nS	k1*nSc7	**k3c4**	k1gInSc7	kIxX	k8	k2gFnSc1d1	k1gFnSc1

Fig. 5: Example comparison between `desamb` and our tagger. Bold attributes are for sure incorrect, other differences can be discussed. The stars mean that our tagger did not provide any attribute and there should be some.

In the near future, we aim to retrain the model with the full tagset. We reduced the tagset mainly because we needed the model to fit into memory. We plan to implement generators to reduce the data needed to be loaded in memory at once.

For future work, we plan to add lemmatizer to the neural network. With the lemmatizer, we want to focus on borrowings and neologisms, since these words are often processed incorrectly by the guesser.

The cleanness of the training data is among known issues, so we plan to solve the problem with incorrect case annotation (mainly nominative and accusative). We will possibly incorporate `majka` outputs and semantic constraints to distinguish these two cases.

Another planned direction is the ability of the tagger to identify foreign injections and annotate MWEs. These two phenomena are related, and we hope

[5] Enter Texas HoldEm Poker into the search field.

[6] The results are published in the journal BMC Biology.

[7] Lady Murasaki was played by Chinese actor Gong Li.

[8] Waters wrote the album The Pros and Cons of Hitch Hiking in 1978 together with The Wall.

[9] Bismuth III sulfide Bi2S3 is dark brown (if prepared by precipitation with hydrogen sulfide) or gray substance.

to solve them using pretrained embeddings, grammatical information, and possibly corpus statistics.

Last but not least, we plan to provide the tagger as a service. We also plan to publish a model based on Universal Dependencies since this widely used tagset will allow a fair evaluation.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin infrastructure LM2015071 and OP VVV project CZ.02.1.01/0.0/0.0/16_013/0001781.

References

1. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
2. Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., Suchomel, V.: The TenTen corpus family. In: 7th International Corpus Linguistics Conference CL 2013. pp. 125–127. Lancaster (2013), http://ucrel.lancs.ac.uk/cl2013/
3. Jakubíček, M., Kovář, V., Šmerk, P.: Czech morphological tagset revisited. In: Horák, R. (ed.) Proceedings of Recent Advances in Slavonic Natural Language Processing 2011. pp. 29–42. Tribun EU, Brno (2011)
4. Pelikánová, Z., Nevěřilová, Z.: czTenTen12 v9 subcorpus of problematic phenomena (2018), http://hdl.handle.net/11234/1-2822, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University
5. Pelikánová, Z., Nevěřilová, Z.: Corpus annotation pipeline for non-standard texts. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) Text, Speech, and Dialogue. pp. 295–303. Springer International Publishing, Cham (2018)
6. Schumacher, M.: Using FastText models (not vectors) for robust embeddings (2 2018), https://www.kaggle.com/mschumacher/using-fasttext-models-for-robust-embeddings
7. Šmerk, P.: Unsupervised learning of rules for morphological disambiguation. In: Sojka, P., Kopeček, I., Pala, K. (eds.) Text, Speech and Dialogue. pp. 211–216. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
8. 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. Association for Computational Linguistics, Baltimore, Maryland (June 2014), http://www.aclweb.org/anthology/P/P14/P14-5003.pdf
9. Suchomel, V.: csTenTen17, a recent czech web corpus. Twelveth Workshop on Recent Advances in Slavonic Natural Language Processing pp. 111–123 (2018)
10. Variš, D., Klyueva, N.: Improving a neural-based tagger for multiword expressions identification. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan (May 2018), https://www.aclweb.org/anthology/L18-1401
11. Šmerk, P.: Fast morphological analysis of Czech. In: Proceedings of the Raslan Workshop 2009. Masarykova univerzita, Brno (2009)