sciendo
DOI 10.2478/jazcas-2023-0049

SLOVAK LANGUAGE MODELS FOR BASIC PREPROCESSING TASKS IN PYTHON

DANIEL HLÁDEK – MAROŠ HARAHUS
– JÁN STAŠ – MATÚŠ PLEVA
Faculty of Electrical Engineering and Informatics, Technical University,
Košice, Slovakia

HLÁDEK, Daniel – HARAHUS, Maroš – STAŠ, Ján – PLEVA, Matúš: Slovak Language Models for Basic Preprocessing Tasks in Python. Journal of Linguistics, 2023, Vol. 74, No 1, pp. 323 – 332.

Abstract: We propose a Slovak language model for the spaCy library in Python. These models are easy-to-use for basic natural language processing tasks in a single package. The package contains several components for basic preprocessing tasks, such as tokenization, sentence boundary detection, syntactic parsing, lemmatization, named entity recognition, morphology analysis, and word vectors. It is based on the state-of-the-art monolingual SlovakBERT model. Named entity recognition is trained on a separate, publicly available WikiAnn database. The other statistical classifiers use a Slovak Dependency Treebank corpus. Morphological tags are compatible with the conventions of the Slovak National Corpus. The part of speech tags use conventions of the Universal Dependencies framework. We trained a separate word vector model on a web-based corpus. The training uses fastText with Floret modification. We present a series of experiments that confirm that the model performs similarly to other languages for all tasks. Training scripts and data are publicly available.

Keywords: Slovak language model, transformers, natural language processing

1 INTRODUCTION

Recently, several models and language resources have appeared for the Slovak language. Thanks to the work of the Slovak National Corpus, a corpus for determining parts of speech (PoS), lemmas (LEM), or dependencies (DEP) (Gajdošová 2016) is available. This dataset was later converted to the Universal Dependency format (Zeman 2017). There are several pre-trained models with the support for the Slovak language, based on the transformer architecture (Devlin 2018; Pikuliak et al. 2022). These new models achieve results comparable to those in other languages. We can say that the basic tasks of natural language processing, such as recognition of parts of speech, are solved and usable for Slovak.

Despite these advances, the available models are fragmented. There are separate models for each task, such as named entity recognition (NER) or part of speech tagging. Compiling a more complex sequence of preprocessing of the Slovak text is

still very laborious and requires detailed knowledge of the available sources. This complicates the development and research of more complex natural language processing tasks, such as dialogue systems, information retrieval, or automatic question answering.

Our models are trained transparently on publicly available data. The training scripts and model download links are available on GitHub.[1] This will allow the feedback collection, easy correction of possible errors, and gradual improvement in the event of the availability of better data or better modeling methods. We use state-of-the-art pretrained models which improve performance, even with smaller training data. This model follows our previous effort (Harahus 2022).

2 STATE OF THE ART

2.1 Available datasets

The most important thing for a statistical model is a good dataset. There are several corpora with morpho-syntactic annotations and named entities available for the Slovak language.

The Slovak Categorized News Corpus (Hládek 2014) is an automatically tagged collection of newspaper articles. This corpus has been meticulously categorized into distinct themes and topics, enabling its use in various natural language processing tasks, such as text classification, sentiment analysis, and topic modeling.

MULTEXT-East (Erjavec 2012) is a multilingual dataset based on the novel "1984" by George Orwell, designed for a variety of natural language processing tasks. It includes resources for several Central and Eastern European languages, such as Bulgarian, Czech, Estonian, Hungarian, Polish, Romanian, Serbian, Slovak, and Slovene. The corpus is annotated with part-of-speech tags, morphological information (MORPH), and lemmatization, making it a valuable resource for researchers and developers working on multilingual natural language processing tasks.

The Slovak Dependency Treebank (Gajdošová 2016) is a syntactically annotated corpus of the Slovak language. The corpus was prepared by the Slovak National Corpus at the Ľ. Štúr Institute of Linguistics of the Slovak Academy of Sciences and its annotation follows the guidelines of the Prague Dependency Treebank (PDT) (Hajič 2017) in Czech. The texts in the treebank come from different genres and domains, such as news articles, literature, and websites. The annotations in the treebank contain information on part-of-speech tags, morphological features, and dependency relations between words in a sentence. This corpus was later converted into Universal Dependencies by Daniel Zeman. The publication (Zeman

[1] https://github.com/hladek/spacy-skmodel

2017) describes the conversion of the syntactically annotated part of the Slovak National Corpus into the Universal Dependencies annotation scheme. The authors selected sentences for which two human annotators agreed 100% on the analysis. The data set is divided into a training set and a test set.

One of the most common datasets for NER training is **WikiAnn**. Cross-lingual Name Tagging and Linking for 282 Languages. Automatic extraction from Wikipedia (WikiAnn) is a large cross-lingual named entity recognition (NER) dataset derived from Wikipedia (Pan 2017). It covers 282 languages, making it a valuable resource for multilingual NLP tasks, especially in the context of NER. The dataset is designed to identify and classify named entities, such as persons, organizations, and places, in a wide range of languages. The annotations in the dataset are in IOB format, which stands for Inside-Outside-Beginning. In this format, each word in a sentence is tagged with a tag that indicates whether it is inside an entity, at the beginning of an entity, or outside of any entity.

2.2 Slovak tools

Tool name	Tasks	Architecture	Language
NLP Cube	SENT, PoS, MORPH, DEP, LEM	CNN	Python
Dl4dp	PoS, DEP, LEM	LSTM	Python
UD Pipe	PoS, MORPH, DEP, LEM	LSTM	C++, Python
Stanza	PoS, MORPH	CNN, LSTM	Python

Tab. 1. Tools for basic natural language preprocessing with Slovak support

There are several tools for preliminary NLP processing of the Slovak language. They are based on convolutional neural networks or long short-term memory without pretraining. Most of them offer a Python interface. They do not contain word vectors and named entity recognition. The tools are summarized in Tab. 1.

NLP-Cube is an end-to-end Natural Language Processing framework that performs sentence splitting, tokenization, compound word expansion, lemmatization, tagging, and parsing. It takes raw-text as input and annotates it, generating a CoNLL-U2 format file. It is written in Python and based entirely on convolutional neural networks built in DyNET2. NLP-Cube is open-source with support for languages included in the UD Treebanks (Boros 2018).

Dl4dp[2] is a Python NLP library that provides tools for morphological tagging, lemmatization, and dependency analysis. It is mainly designed for English but can also work with other languages that have training data in universal dependencies. Universal Dependencies is a framework for consistent annotation of grammar across languages.

[2] https://github.com/peterbednar/dl4dp

UDPipe is a trainable pipeline for tokenization, tagging, lemmatization, and dependency analysis of CoNLL-U files. It is not language dependent and can work with any language that has annotated data in CoNLL-U format. CoNLL-U is a format for consistent annotation of grammar across languages. UDPipe is available as a library for C++, Python and R (Straka 2016).

Stanza is an NLP library for Python languages. It supports many human languages. It provides pre-trained models for various NLP tasks, such as part-of-speech tagging, named entity recognition, dependency analysis, and sentiment analysis. It also includes tokenization features, sentence splitting, and multi-word token expansion (Qui 2020).

2.3 SpaCy framework

SpaCy[3] features state-of-the-art models for tasks such as named entity recognition, part-of-speech tagging, and dependency parsing, as well as pre-trained models for several languages. With its efficient processing speed and intuitive API, spaCy makes it easy foS developers to incorporate NLP capabilities into their applications. SpaCy supports several languages, including German, Spanish, French, Italian, Portuguese, Dutch, Greek, Norwegian, Lithuanian, Danish, Finnish, Hungarian, Romanian, Slovenian, and Swedish. The classification models are based on a deep learning approach known as transformers (Devlin 2017), which is used in state-of-the-art natural language processing (NLP) models. In the paper (Ye 2019) authors investigate the use of multitask learning and word embeddings for named entity recognition in informal language texts. The authors use spaCy and several reference datasets for their experiments.

3 THE PROPOSED ARCHITECTURE

Training a custom model in spaCy involves defining the pipeline components, specifying the training data, and selecting the hyperparameters for training. The training pipeline is a sequence of components. Components are applied to a text document to extract useful information from it. The default pipeline components include tokenization, part-of-speech tagging, lemmatization, dependency parsing, and named entity recognition. Besides that, there is a word vector component that is available upon request.

The tokenizer takes a text string and splits it into individual tokens. The tokenizer in spaCy is rule-based; it splits the text based on whitespace and punctuation marks and processes special cases such as abbreviations and acronym words.

Part-of-speech tagging, lemmatization, and dependency parsing are performed using the first neural model trained on the UD Slovak Dependency treebank (Zeman

[3] https://spacy.io/

2017). Named entity recognition is performed using another neural model, trained on a WikiAnn (Pan 2017) corpus. Both neural models use a pretrained Slovak model based on a RoBERTa architecture (Liu et al. 2019), called SlovakBERT (Pikuliak et al. 2022). The SlovakBERT model is trained on a large corpus of Slovak text, leveraging the power of the transformer architecture and the masked language modeling objective.

3.1 Word vectors

Word vectors represent each word in the dictionary as a high-dimensional vector, where words that are semantically similar to each other are close to each other in vector space (Mikolov 2013). The word vectors module uses a modification of the version of fastText (Joulin 2016). called Floret. This modification uses Bloom inserts to create compact vector tables of word information. Floret vectors are trained using a technique called "sliced sparse coding", which helps to reduce the dimensionality of the vectors while maintaining their precision. This makes them much smaller and faster to compute than traditional nested words, which can be important for applications where computational resources are limited, such as mobile devices or embedded systems.

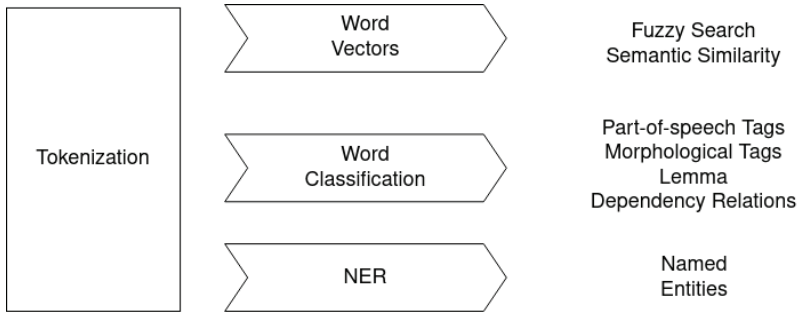


Fig. 1. The proposed model architecture

3.2 Word classification model

This model uses the Transformers architecture (SlovakBERT) and is independent of other models. It performs:

1. POS tagging (into UD tags);
2. morphological analysis (into SNK tags);
3. dependency parsing (into UD tags);
4. lemmatization.

The role of the PoS tagger is to take the output of the tokenizer and assign a PoS tag to each token based on its context in the sentence. It uses contextual information such as the surrounding words and the grammatical structure of the

sentence. It tries to predict the most likely PoS tag for each token. The POS tags follow the conventions of the Universal Dependencies corpora. The output is a sequence of token objects that have been annotated with the corresponding PoS tags (Partalidou 2019). Morphological analysis refers to the more precise analysis of a specific word form. We use the Slovak National Corpus tagset[4] for this purpose. The output of a dependency parser is a tree structure that represents the dependency relations between words in a sentence (Colic 2019). The dependency tags are the same as in the training corpus.

3.3 Named entity recognizer

The second neural model performs named entity recognition. This model uses the transformer architecture and is independent of other models. The model is trained separately because the training corpus is different and has a different distribution license.

The output of the named entity recognizer is a document that contains the identified named entities and their corresponding tags. A named entity is a non-overlapping span in the document that contains one or multiple words. The named entity recognizer assigns three different tags to each word – beginning, inside, outside (BIO). The tags mark whether a word is the beginning of the named entity (B), is inside it (O), or does not belong to a named entity (O). Each word in the input sequence has exactly one tag assigned; thus, the named entities cannot overlap.

4 TRAINING AND EVALUATION

SpaCy offers several different pre-trained model architectures that are optimized for different natural language processing tasks. The „small" model architecture is a basic NLP pipeline that includes tokenization, part-of-speech tagging, and dependency parsing. The „medium" model architecture adds named entity recognition to the basic pipeline of the small model. The „large" model architecture includes all the components of the medium model as well as a deep neural network that is trained on a large corpus of textual data. Our models contain PoS, DEP, and NER taggers, so they are marked as „medium".

4.1 The training process

We trained three statistical models for spaCy:

1. Word vectors – modified fastText on custom web corpus;
2. Word classifier – SlovakBERT with Slovak Dependency Treebank UD corpus;
3. NER recognizer – SlovakBERT with WikiAnn NER corpus.

[4] https://www.juls.savba.sk/~vladob/resources/20180403_ukl_cheatsheet_0.pdf

The workstation used for this experiment was a standalone Ubuntu-Linux PC with a 2.67 GHz Core i7920 processor, 32GB of RAM, and 2 NVIDIA GeForce 1080 graphics cards with 12 GB of VRAM each.

In the first step, we trained word vectors. The word vectors were trained on a large Slovak web-based corpus using a modified fastText tool.[5] We gathered many the Slovak web pages from the Internet. The text was extracted using jusText[6] and tokenized by a rule-based tokenizer.[7] Redundancy was lowered by detecting duplicities on the level of paragraphs. The training corpus has over 4 billion tokens. We trained the model with the recommended hyperparameters: method CBOW/floret, dimension 300, minn 4, maxn 5, hash count 2, bucket 50000.

Then we trained the word classifier and named entity recognizer separately. For both models, we used pretrained SlovakBERT. Hyperparameters were the same for both models: learning rate 1e-5, update freq. 3, dropout 0.1, max. steps 2000, optimizer Adam.

During training we monitored these metrics:

- Part-of-speech accuracy (PoS) and morphological analysis accuracy (MORPH);
- Lemmatizer Accuracy (LEM);
- Unlabeled dependencies score (UAS): which measures the percentage of tokens whose head is correctly predicted;
- Labeled dependencies (LAS): measures the percentage of tokens whose head and dependency label are correctly predicted;
- Named entity recognition precision, recall, and F1.

4.2 The evaluation

The scores were calculated on a development set, left over from the training. The development set consists of 10% of the total data. Results of word classification training are displayed in Tab. 2. We added results for English, German, and Polish languages for comparison. Results for other languages were recorded from the spaCy webpage for "core_medium" models.

Language	PoS	MORPH	LEM	DEP_UAS	DEP_LAS
English	0.97	n-a	n-a	0.92	0.90
German	0.92	0.92	0.98	0.93	0.91
Polish	0.98	0.90	0.94	0.89	0.81
Slovak	0.92	0.94	0.86	0.92	0.89

Tab. 2. Results of the Slovak word classification, compared with other languages

[5] https://github.com/explosion/floret

[6] https://github.com/miso-belica/jusText

[7] https://github.com/hladek/slovak-lexer

The results of the named entity recognition model training are in separate Tab. 3, where the F1, precision, and recall scores for each language are presented.

Language	F1	Precision	Recall
EN	0.85	0.85	0.85
DE	0.84	0.84	0.83
PL	0.83	0.83	0.82
SK	0.89	0.89	0.89

Tab. 3. Results of the named entity recognition

The tables show that the results are comparable with other languages. The only significant difference drop of precision was for the Slovak lemmatization when compared to Polish and German. The reason requires further investigation.

5 CONCLUSION

In this work, we solve the problem of fragmentation and provide a complete pipeline for the preprocessing of the Slovak language using state-of-the-art models. Our approach uses proven means – the Python programming language and the spaCy library with transformers architecture. With one command,[8] it is possible to install a Slovak language preprocessing tool for tokenization, recognition of parts of speech and lemmas, parsing, and recognition of named entities. The experiments have shown that the results are comparable with other languages. This will facilitate access to NLP technologies even for beginners and programmers unfamiliar with the Slovak NLP. The spaCy library already provides support for several other languages, so it is possible to easily add support for the Slovak language as well as for existing applications.

ACKNOWLEDGEMENTS

The research in this paper was supported by the Scientific Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic and the Slovak Academy of Sciences under the project VEGA 2/0165/21, and by the Slovak Research and Development Agency under the projects APVV-22-0261, APVV-SK-TW-21-0002, and APVV-22-0414.

References

Boroş, T., Dumitrescu, S. D., and Burtica, R. (2018). NLP-Cube: End-to-end raw text processing with neural networks. In Proc. of the CoNLL 2018 Shared Task: Multilingual

[8] pip install https://files.kemt.fei.tuke.sk/models/spacy/sk_core_web_md-3.4.1.tar.gz

Parsing from Raw Text to Universal Dependencies, Brussels, Belgium, pages 171–179. Accessible at: https://aclanthology.org/K18-2017.pdf.

Colic, N., and Rinaldi, F. (2019). Improving spaCy dependency annotation and PoS tagging web service using independent NER services. Genomics Inform., 17(2) e21. Accessible at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6808626/.

Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proc. of NAACL HLT, Minneapolis, Minnesota, pages 4171–4186. Accessible at: https://aclanthology.org/N19-1423.pdf.

Erjavec, T. (2012). MULTEXT-East: morphosyntactic resources for Central and Eastern European languages Language Resources and Evaluation, 46(1), pages 131–142. Accessible at: https://www.jstor.org/stable/41486069.

Gajdošová, K., Šimková, M. et al. (2016). Slovak dependency treebank. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. Accessible at: https://lindat.cz/repository/xmlui/handle/11234/1-1822.

Hajič, J., Hajičová, E., Mikulová, M., and Mírovský, J. (2017). Prague dependency treebank. In Handbook of Linguistic Annotation, pages 555–594.

Harahus, M, Juhár, J., and Hládek D. (2022). Morphological annotation of the Slovak language in the Spacy library with the pretraining. 32nd International Conference Radioelektronika (RADIOELEKTRONIKA). IEEE, 2022. Accessible at: doi 10.1109/RADIOELEKTRONIKA54537.2022.9764935.

Hládek, D., Staš, J., and Juhár, J. (2014). The Slovak Categorized News Corpus. In LREC, pages 1705–1708. Accessible at: https://aclanthology.org/L14-1517/.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. In Proc. of EACL: Volume 2, Short Papers, Valencia, Spain, pages 427–431. Accessible at: https://aclanthology.org/E17-2068.pdf.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., and Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. CoRR, arXiv preprint, arXiv:1907.11692. Accessible at: https://arxiv.org/pdf/1907.11692.pdf.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. CoRR, arXiv preprint, arXiv:1301.3781. Accessible at: https://arxiv.org/pdf/1301.3781.pdf.

Pan, X., Zhang, B., May, J., Nothman, J., Knight, K., and Ji, H. (2017). Cross-lingual name tagging and linking for 282 languages. In Proc. of ACL: Volume 1, Long Papers, Vancouver, Canada, pages 1946–1958. Accessible at: https://aclanthology.org/P17-1178.pdf.

Partalidou, E., Spyromitros-Xioufis, E., Doropoulos, S., Vologiannidis, S., and Diamantaras, K. (2019). Design and implementation of an open source Greek PoS tagger and entity recognizer using spaCy. In Proc. of WI'19: IEEE/WIC/ACM International Conference on Web Intelligence, Thessaloniki, Greece, pages 337–341. Accessible at: https://dl.acm.org/doi/10.1145/3350546.3352543.

Pikuliak, M., Grivalský, Š., Konôpka, M., Blšták, M., Tamajka, M., Bachratý, V., Šimko, M., Balážik, P., Trnka, M., and Uhlárik, F. (2022). SlovakBERT: Slovak masked language model. In Proc. of EMNLP, Abu Dhabi, United Arab Emirates, pages 7156–7168. Accessible at: https://aclanthology.org/2022.findings-emnlp.530.pdf.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, D. Ch. (2020). Stanza: A Python natural language processing toolkit for many human languages. In Proc. of ACL: System Demonstrations, Online, pages 101–108. Accessible at: https://aclanthology.org/2020.acl-demos.14.pdf.

Straka, M., Hajič, J., and Straková, J. (2016). UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, PoS tagging and parsing. In Proc. of LREC, Portorož, Slovenia, pages 4290–4297. Accessible at: https://aclanthology.org/L16-1680.pdf.

Ye, W., Li, B., Xie, R., Sheng, Z., Chen, L., and Zhang, S. (2019). Exploiting entity BIO tag embeddings and multi-task learning for relation extraction with imbalanced data. arXiv preprint arXiv:1906.08931.

Zeman, D. (2017). Slovak dependency treebank in universal dependencies. Jazykovedný časopis, 68(2), pages 385–395. Accessible at: https://sciendo.com/article/10.1515/jazcas-2017-0048.