SlovakBERT: Slovak Masked Language Model

Anonymous ACL submission

Abstract

We introduce a new Slovak masked language model called *SlovakBERT*. This is to our best knowledge the first paper discussing Slovak transformers-based language models. We evaluate our model on several NLP tasks and achieve state-of-the-art results. This evaluation is likewise the first attempt to establish a benchmark for Slovak language models. We publish the masked language model, as well as the fine-tuned models for part-of-speech tagging, sentiment analysis and semantic textual similarity.

1 Introduction

Fine-tuning pre-trained large-scale language models (LMs) is the dominant paradigm of current NLP. LMs proved to be a versatile technology that can help to improve performance for an array of NLP tasks, such as parsing, machine translation, text summarization, sentiment analysis, semantic similarity etc. The state-of-the-art performance makes LMs attractive for any language community that wants to develop their NLP capabilities. In this paper, we concern ourselves with Slovak language and address the lack of language models, as well as the lack of established evaluation standards for this language.

In this paper, we introduce a new Slovak-only transformers-based language model called *SlovakBERT*[1]. Although several multilingual models already support Slovak, we believe that developing Slovak-only models is still important, as it can lead to better results and more compute and memory-wise efficient processing of Slovak language. *SlovakBERT* has RoBERTa architecture (Liu et al., 2019) and it was trained with a Web-crawled corpus.

Since no standard evaluation benchmark for Slovak exists, we created our own set of tests mainly from pre-existing datasets. We believe that our evaluation methodology might serve as a standard benchmark for Slovak language in the future. We evaluate *SlovakBERT* with this benchmark and we also compare it to other available (mainly multilingual) LMs and other existing approaches. The tasks we use for evaluation are: part-of-speech tagging, semantic textual similarity, sentiment analysis and document classification. We also publish the best performing models for selected tasks. These might be used by other Slovak researchers or NLP practitioners in the future as strong baselines.

Our main contributions in this paper are:

- We published a Slovak-only LM trained on a Web corpus.
- We established an evaluation methodology for Slovak language and we apply it on our model, as well as on other LMs.
- We published several fine-tuned models based on our LM, namely a part-of-speech tagger, a sentiment analysis model and a sentence embedding model.
- We published several additional datasets for multiple tasks, namely sentiment analysis test sets and semantic similarity translated dataset.

The rest of this paper is structured as follows: In Section 2 we discuss related work about language models and their language mutations. In Section 3 we describe the corpus crawling efforts and how we train *SlovakBERT* with the resulting corpus. In Section 4 we evaluate the model with four NLP tasks.

2 Related Work

2.1 Language Models

LMs today are commonly based on self-attention layers called *transformers* (Vaswani et al., 2017). Despite the common architecture, the models might differ in the details of their implementation, as well

[1] Available at `https://github.com/...`

as in the task they are trained with (Xia et al., 2020). Perhaps the most common task is the so called *masked language modeling* (Devlin et al., 2019a), where randomly selected parts of text are masked and the model is expected to fill these parts with the original tokens. Masked language models are useful mainly as backbones for further fine-tuning. Another approach is to train a generative autoregressive models (Radford et al., 2019), that always predicts the next word in a sequence, which can be used for various text generation tasks. Variants of LMs exist that attempt to make them more efficient (Clark et al., 2020; Jiao et al., 2020), able to handle longer sentences (Beltagy et al., 2020) or fulfill various other requirements.

2.2 Availability in Different Languages

English is the most commonly used language in NLP, and a *de facto* standard for experimental work. Most of the proposed LM variants are indeed trained and evaluated only on English. Other languages usually have at most only a few LMs trained, usually with a very safe choice of model architecture (e.g. BERT or RoBERTa). Languages with available native models are e.g. French (Martin et al., 2020), Dutch (Delobelle et al., 2020) or Arabic (Antoun et al., 2020). There are also models for related Slavic languages, notably Czech (Sido et al., 2021) and Polish (Dadas et al., 2020).

There is no Slovak-specific large scale LM available so far. There is a Slovak version of WikiBERT model (Pyysalo et al., 2021), but it is trained only on texts from Wikipedia, which is not a large enough corpus for proper language modeling at this scale. The limitations of this model will be shown in the results as well.

2.3 Multilingual Language Models

Multilingual LMs are sometimes proposed as an alternative to training language-specific LMs. These LMs can handle more than one language. In practice, they are often trained with more than 100 languages. Training them is more efficient than training separate models for all the languages. Additionally, cross-lingual transfer learning might improve the performance with the languages being able to learn from each other. This is especially beneficial for low-resource languages.

The first large-scale multilingual LM is MBERT (Devlin et al., 2019a) trained on 104 languages. The authors observed that by simply exposing the model to data from multiple languages, the model was able to discover the multilingual signal and it spontaneously developed interesting cross-lingual capabilities, i.e. sentences from different languages with similar meaning also have similar representations. Other models explicitly use multilingual supervision, e.g. dictionaries, parallel corpora or machine translation systems (Conneau and Lample, 2019; Huang et al., 2019)

3 Training

In this section we describe our own Slovak masked language model – *SlovakBERT*, the data that were used for training, the architecture of the model and how it was trained.

3.1 Data

We used a combination of available corpora and our own Web-crawled corpus as our training data. The available corpora we used were: Wikipedia (326MB of text), Open Subtitles (415MB) and OSCAR corpus (4.6GB). We crawled `.sk` top-level domain webpages, applied language detection and extracted the title and the main content of each page as clean text without HTML tags (17.4GB). The text was then processed with the following steps:

- URL and email addresses were replaced with special tokens.
- Elongated interpunction was reduced, i.e. if there were sequences of the same interpunction character, these were reduced to one character (e.g. `--` to `-`).
- Markdown syntax was deleted.
- All text content in braces `{.}` was eliminated to reduce the amount of markup and programming language text.

We segmented the resulting corpus into sentences and removed duplicates to get 181.6M unique sentences. In total, the final corpus has 19.35GB of text.

3.2 Model Architecture and Training

The model itself is a RoBERTa model (Liu et al., 2019). The details of the architecture are shown in Table 1 in the `SlovakBERT` column. We use BPE (Sennrich et al., 2016) tokenizer with the vocabulary size of 50264. The model was trained for 300k training steps with a batch size of 512. Samples were limited to a maximum of 512 tokens and for each sample we fit as many full sentences as possible. We used Adam optimization algorithm (Kingma and Ba, 2015) with 5×10^{-4} learning rate and 10k warmup steps. Dropout (dropout rate 0.1) and weight decay ($\lambda = 0.01$) were used for regularization. We used `fairseq` (Ott et al., 2019) library for training, which took approximately 248 hours on 4 NVIDIA A100 GPUs. We used 16-bit float precision.

4 Evaluation

In this section, we describe the evaluation methodology and results for *SlovakBERT* and other LMs. We use two main methods to examine the performance of LMs:

1. *Fine-tuned performance.* We fine-tune the LMs for various NLP tasks and we analyze the achieved results. We compare the results with existing solutions based on other approaches, e.g. rule-based solutions or solutions based on word embeddings.

2. *Probing.* Probing is a technique that aims to measure the amount of relevant information on individual layers of LMs. We use simple *linear probes* in our work, i.e. the hidden representations from the LMs are used as features for linear classifiers.

We conducted the evaluation on four different tasks: part-of-speech tagging, semantic textual similarity, sentiment analysis and document classification. For each task, we introduce the dataset that is used, various baselines solutions, the LM-based approach we took and the final results for the task.

4.1 Evaluated Language Models

We evaluate and compare several LMs that support Slovak language to some extent:

XLM-R (Conneau et al., 2020) - XLM-R is a suite of multilingual RoBERTa-style LMs. The models support 100 languages, including Slovak. Training data are based on CommonCrawl Web-crawled corpus. Slovak part has 23.2 GB (3.5B tokens). The XLM-R models differ in their size, ranging from Base model with 270M parameters to XXL model with 10.7B parameters.

MBERT (Devlin et al., 2019b) - MBERT is a multilingual version of the original BERT model trained with Wikipedia-based corpus containing 104 languages. Authors do not mention the amount of data for each language, but considering the size of Slovak Wikipedia, we assume that the Slovak part has tens of millions of tokens.

WikiBERT (Pyysalo et al., 2021) - WikiBERT is a series of monolingual BERT-style models trained on dumps of Wikipedia. The Slovak model was trained with 39M tokens.

Note that both XLM-R and MBERT models were trained in cross-lingually unsupervised manner, i.e. no additional signal about how sentences or words from different languages relate to each other was provided. The models were trained with a multilingual corpora only, although language balancing was performed.

In Table 1 we provide a basic quantitative measures for all the models. We compare their architecture and training data. We also measure tokenization productivity on texts from *Universal Dependencies* (Nivre et al., 2020) train set. We show the average length of tokens for each model. Longer tokens are considered to be better, because they can be more semantically meaningful and also because they are more computationally efficient. We also show how many unique tokens were used (effective vocabulary) for the tokenization of this particular dataset. Multilingual LMs have smaller portion of their vocabulary used, since they contain many tokens useful mainly for other languages, but not for Slovak. These tokens are effectively redundant for Slovak text processing.

4.2 Part-of-Speech Tagging

The goal of part-of-speech (POS) tagging is to assign a certain POS tag from the predefined set of possible tags to each word. This task mainly evaluates the syntactic capabilities of the models.

4.2.1 Data

We use Slovak Dependency Treebank from *Universal Dependencies* dataset (Zeman, 2017; Nivre et al., 2020) (UD). It contains annotations for both Universal (UPOS, 17 tags) and Slovak-specific (XPOS, 19 tags) POS tagsets. XPOS uses a more complicated system and it encodes not only POS tags, but also other morphological categories in the label. In this work, we only use the first letter from each XPOS label, which corresponds to a typical POS tag. The tagsets and their relations are shown in Table 8.

Model	SlovakBERT	XLM-R-Base	XLM-R-Large	MBERT	WikiBERT
Architecture	RoBERTa	RoBERTa		BERT	BERT
Num. layers	12	12	24	12	12
Num. attention head	12	12	16	12	12
Hidden size	768	768	1024	768	768
Num. parameters	125M	278M	560M	178M	102M
Languages	1	100	100	104	1
Training dataset size (tokens)	4.6B	167B		n/a	39M
Slovak dataset size (tokens)	4.6B	3.2B		25-50M	39M
Vocabulary size	50K	250K		120K	20K
Average token length *	3.23	2.84		2.40	2.70
Effective vocabulary *	16.6K	9.6K		6.7K	5.8K
Effective vocabulary (%) *	33.05	3.86		5.62	29.10

Table 1: Basic statistics about the evaluated LMs. *Data are calculated based on *Universal Dependencies* dataset.

4.2.2 Previous work

Since Slovak is an official part of the UD dataset, systems that attempt to cover multiple or all UD languages often support Slovak as well. The following systems were trained on UD data and support both UPOS and XPOS tagsets:

UDPipe 2 (Straka, 2018) - A deep learning model based on multilayer bidirectional LSTM architecture with pre-trained Slovak word embeddings. The model supports multiple languages, but the models themselves are monolingual.

Stanza (Qi et al., 2020) - Stanza is a very similar model to UDPipe, it is also based on multilayer bidirectional LSTM with pre-trained word embeddings.

Trankit (Nguyen et al., 2021) - Trankit is based on adapter-style fine-tuning (Bapna and Firat, 2019) of XLM-R-Base. The adapters are fine-tuned for specific languages and they are able to handle multiple tasks at the same time.

4.2.3 Our Fine-Tuning

We use a standard setup for fine-tuning the LMs for token classification. The final layer of an LM that is used to predict the masked tokens is discarded. A classifier linear layer with dropout is used in its place to generate POS tag logits for each token. These logits are then transformed to a probability vector with softmax function and a cross-entropy is calculated for each token. The loss function for batch of samples is defined as an average cross-entropy across all the tokens. For inference, we simply pick the class with the highest probability for each token. Note that there is a discrepancy between what we perceive as words and what the models use as tokens. Some words might be tokenized into multiple tokens. In that case, we only make the prediction on the first token and the final classifier layer is not applied to the subsequent tokens for this word. We use `Hugging Face Transformers` library for LM fine-tuning.

We use similar setup for probing, but with two changes: (1) We freeze all the weights apart from the classifier layer, and (2) we remove several top layers from the LM, i.e. instead of making predictions from the topmost layer, we make them from other layer instead. This way we can analyze how well the representations generated on each layer work.

4.2.4 Results

We have performed a random hyperparameter search with *SlovakBERT*. The range of individual hyperparameters is shown in Table 6. We have found out that weight decay is a beneficial regularization technique, while label smoothing proved itself to be inappropriate for our case. Other hyperparameters showed to have a very little reliable effect, apart from learning rate, which proved to be very sensitive. We have not repeated this tuning for other LMs, instead, we only tuned the learning rate. We have found out that it is appropriate to use learning rate of 1×10^{-5} for all the models, but XLM-R-Large. XLM-R-Large, the biggest model we tested, needs smaller learning rate of 1×10^{-6}.

The results for POS tagging are shown in Table 2. We report accuracy for both XPOS and UPOS tagsets. WikiBERT seems to be the worst-performing LM, probably because of its small training set. *SlovakBERT* seems to be on par with larger XLM-R-Large. Other models lag behind slightly. From existing solutions, only transformers-based Trankit seems to be able to keep up.

We also analyzed the dynamics of the LM fine-tuning. We analyzed the performance for various

Model	UPOS	XPOS
UDPipe 2.0	92.83	94.74
UDPipe 2.6	97.30	97.87
Stanza	96.03	97.29
Trankit	97.85	98.03
WikiBERT	94.41	96.54
MBERT	97.50	98.03
XLM-R-Base	97.61	98.23
XLM-R-Large	**97.96**	98.34
SlovakBERT	97.84	**98.37**

Table 2: Results for POS tagging (accuracy).

checkpoints of our LM (checkpoints were made after 1000 training steps). We can see in Figure 1, that *SlovakBERT* was saturated w.r.t POS performance quite soon, after approximately 15k steps. We stopped the analysis after the first 125k steps, since the results seemed to be stable. Similar results for probing can be seen in the same figure. We show the performance for all the layers for selected checkpoints. The performance on layers peaks quite soon at layer 6 and then plateaus. The last layers even have degraded performance. This shows, that the morphosyntactic information needed for POS tagging is stored and processed mainly in the middle part of the model. This is in accord with the current knowledge about how LMs work, i.e. that they process the text in a bottom-up manner (Tenney et al., 2019).

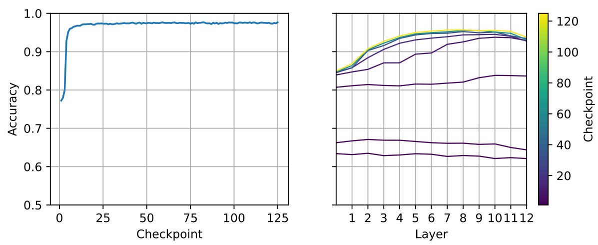

Figure 1: Analysis of POS tagging learning dynamics. *Left:* Accuracy after fine-tuning the different checkpoints. *Right:* Accuracy of probes on all the layers of different checkpoints.

4.3 Semantic Textual Similarity

Semantic textual similarity (STS) is an NLP task where a similarity between pairs of sentences is measured. In our work, we train the LMs to generate sentence embeddings and then we measure how much the cosine similarity between embeddings correlates with the ground truth labels provided by human annotators. We can use the resulting models to generate universal sentence embeddings for Slovak.

4.3.1 Data

Currently, there is no native Slovak STS dataset. We decided to machine translate existing English datasets, namely STSbenchmark (Cer et al., 2017) and SICK (Marelli et al., 2014) into Slovak. These datasets use a $\langle 0, 5 \rangle$ scale that expresses the similarity of two sentences. The meaning of individual steps on this scale is shown in Table 9. English STS systems also usually use natural language inference (NLI) data to perform additional pre-training. NLI is a task where the goal is to identify cases of entailment or contradiction between two sentences. We translated SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) datasets to Slovak as well. We use M2M100 (1.2B parameters variant) machine translation system (Fan et al., 2021).

4.3.2 Previous Work

No Slovak-specific sentence embedding model has been published yet. We use a naive solution based on Slovak word embeddings and several available multilingual models for comparison:

fastText (Bojanowski et al., 2017) - We use pre-trained Slovak fastText word embeddings to generate representations for individual words. The sentence representation is an average of all its words. This represents a very naive baseline, since it completely omits the word order.

LASER (Artetxe and Schwenk, 2019) - LASER is a model trained to generate multilingual sentence embeddings. It is based on an encoder-decoder LSTM machine translation system that is trained with 93 languages. The encoder is shared across all the languages and as such, it is able to generate multilingual representations.

LaBSE (Feng et al., 2020) - LaBSE is an MBERT model fine-tuned with parallel corpus to produce mutlilingual sentence representations.

XLM-R$_{EN}$ (Reimers and Gurevych, 2020) - XLM-R model fine-tuned with English STS-related data (SNLI, MNLI and STSbenchmark datasets). This is a zero-shot cross-lingual learning setup, i.e. no Slovak data are used and only English fine-tuning is done.

4.3.3 Our Fine-Tuning

We use a setup similar to (Reimers and Gurevych, 2020). A pre-trained LM is used to initialize a

Model	Score
fastText	0.383
LASER	0.711
LaBSE	0.739
XLM-R$_{EN}$	**0.801**
WikiBERT	0.673
MBERT	0.748
XLM-R-Base	0.767
XLM-R-Large	0.791
SlovakBERT	0.791

Table 3: Spearman correlation between cosine similarity of generated representations and desired similarities on STSbenchmark dataset translated to Slovak.

Siamese network. Both branches of the network are identical LMs with a mean-pooling layer at the top that generates the final sentence embeddings. The embeddings from the two sentences are compared using cosine similarity. The network is trained as a regression model, i.e. the final computed similarity is compared with the ground truth similarity with *mean squared error* loss function. We use `SentenceTransformers` library for the fine-tuning.

We also performed a layer-wise analysis, where we analyzed which layers have the most viable representations for this task. We conducted the mean-pooling at different layers and ignored all the subsequent layers. This is similar to probing, but probing is usually done with frozen LM layers. In this case, we can not freeze the layers, since all the additional layers we added (mean-pooling, cosine similarity calculation) are not parametric.

4.3.4 Results

We compare the systems using Spearman correlation between the cosine similarity of the generated sentence representations and the ground truth data. The original STS datasets are using $\langle 0, 5\rangle$ scale. We normalize these scores to $\langle 0, 1\rangle$ range so that they can be directly compared to the cosine similarities. We performed a hyperparameter search in this case as well. Again, we have found out that the results are quite stable across various hyperparameter values, with learning rate being the most sensitive hyperparameter. The details of the hyperparameter tuning are shown in Table 7. We show the main results in Table 3.

We can see that the results are fairly similar to POS tagging w.r.t. how the LMs are relatively ordered. The existing solutions are worse, except for XLM-R$_{EN}$ trained with English data, which is actually the best performing model in our experiments. It seems that their model fine-tuned with real data without machine-translation-induced noise works better, even if it has to perform the inference cross-lingually on Slovak data.

We also experimented with Slovak-translated NLI data in a way where the model was first fine-tuned on NLI task and then the final STS fine-tuning was performed. However, we were not able to outperform the purely STS fine-tuning with this approach and the results remained virtually the same. This result is in contrast with the usual case for English training, where the NLI data regularly improve the results (Reimers and Gurevych, 2019). We theorize that this effect might be caused by noisy machine translation.

Figure 2 shows the learning dynamics of STS. On the left, we can see that the performance takes much longer to plateau than in the case of POS. This shows that the model needs longer time to learn about semantics. Still, we can see that the performance ultimately stabilizes just below 0.8 score. Similarly, unlike POS, we can see that the best performing layers are actually the last layers of the model.

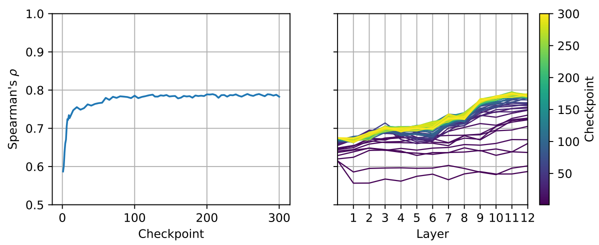

Figure 2: Analysis of STS learning dynamics. *Left:* Spearman correlation after fine-tuning with various checkpoints. *Right:* Spearman correlation on all the layers with selected checkpoints.

4.4 Sentiment Analysis

The goal of sentiment analysis is to identify the affective sentiment of a given text. It requires semantic analysis of the text, as well as certain amount of emotional understanding.

4.4.1 Data

We use a Twitter-based dataset (Mozetič et al., 2016) annotated on a scale with three values: *negative*, *neutral* and *positive*. Some of the tweets have already been removed since the dataset was created.

Therefore, we work with a subset of the original dataset.

We cleaned the data by removing URLs, retweet prefixes, hashtags, user mentions, quotes, asterisks, redundant whitespaces and trailing punctuation. We have also deduplicated the samples, as there were cases of identical samples (i.e. retweets) or very similar samples (i.e. automatically generated tweets). These duplicates had in some cases different labels. After the deduplication, we were left with 41084 tweets with 11160 negative samples, 6668 neutral samples and 23256 positive samples.

Additionally, we have also manually annotated a series of test sets containing reviews from various domains: accommodation, books, cars, games, mobiles and movies. Each domain has approximately 100 manually labeled samples. These are published along with this paper. They serve to check how well the model behavior transfers to other domains. This dataset is called *Reviews* in the results below, while the original Twitter-based dataset is called *Twitter*.

4.4.2 Previous Work and Baselines

The original paper introducing the Twitter dataset introduced an array of traditional classifiers (Naive Bayes and 5 SVM variants) to solve the task. The authors report the macro-F1 score for positive and negative classes only. Additionally, unlike us, they worked with the whole dataset. Approximately 10K tweets have been deleted since the dataset was introduced. (Pecar et al., 2019) use the same version of the dataset as we do. They use approaches based on word embeddings and ELMO (Peters et al., 2018) to solve the task. Note that both published works use cross-validation, but no canonical dataset split is provided in either of them.

There are several existing approaches we use for comparison:

NLP4SK[2] - A rule-based sentiment analysis system for Slovak that is available online

Amazon - We also translated the Slovak data into English and used Amazon's commercial sentiment analysis API and tested its performance on our test sets.

We implemented several baseline classifiers that were trained with the same training data as the LMs in our experiments:

[2] `http://arl6.library.sk/nlp4sk/webapi/analyza-sentimentu`

TF-IDF linear classifier - A perceptron trained with SGD algorithm. The text is represented with TF-IDF using N-grams as basic text units.

fastText classifier - We used the built-in fastText classifier with and without pre-trained Slovak word embedding models.

Our STS embedding linear classifier - A perceptron trained with SGD algorithm. The text is represented using the sentence embedding model we have trained for STS.

We performed a random search hyperparameter optimization for all the approaches.

4.4.3 Our Fine-Tuning

We fine-tuned the LMs as classifiers with 3 classes. The topmost layer of an LM is discarded and instead a multilayer perceptron classifier with one hidden layer and dropout is applied on the representation of the first token. Categorical cross-entropy loss function is used as loss function. The class with the highest probability coming from the softmax function is selected as the predicted label during inference. We use `Hugging Face Transformers` library for fine-tuning.

4.4.4 Results

We report macro-F1 scores for all three classes as our main performance measure. The LMs were trained on the Twitter dataset. We calculate average F1 from our *Reviews* dataset as an additional measure.

Again, we have performed a hyperparameter optimization of *SlovakBERT*. The results are similar to results from POS tagging and STS. We have found out that learning rate is the most sensitive hyperparameter and that a small amount of weight decay is a beneficial regularization. The main results are shown in Table 4. We can see that we were able to obtain better results than the results that were reported previously. However, the comparison is not perfect, as we use slightly different datasets for the aforementioned reasons.

The LMs are ordered in performance similarly to how they are ordered in the two previous tasks. *SlovakBERT* seems to be among the best performing models, along with the larger XLM-R-Large. The LMs were also able to successfully transfer their sentiment knowledge to new domains and they achieve up to 0.617 macro-F1 in the reviews

Model	Twitter F1		Reviews F1
	3-class	2-class	3-class
(Mozetič et al., 2016)*	-	0.682	-
(Pecar et al., 2019)*	0.669	-	-
Amazon	0.502	0.472	0.766
NLP4SK	0.489	0.468	**0.815**
TF-IDF	0.571	0.603	0.412
fastText	0.591	0.622	0.416
fastText w/ emb.	0.606	0.631	0.426
STS embeddings	0.581	0.597	0.582
WikiBERT	0.580	0.597	0.398
MBERT	0.587	0.622	0.453
XLM-R-Base	0.620	0.651	0.518
XLM-R-Large	0.655	**0.716**	0.617
SlovakBERT	**0.672**	0.705	0.583

Table 4: Macro-F1 scores for sentiment analysis task. The 2-class F1 score for Twitter is calculated only from positive and negative classes – a methodology introduced in the original dataset paper. *Indicates different evaluation sets.

as well. However, both Amazon commercial sentiment API and NLP4SK have even better scores, even though their performance on Twitter data was not very impressive. This is probably caused by the underlying training data they use in their systems, that might match our *Reviews* datasets more than the tweets used for our fine-tuning.

4.5 Document Classification

The final task which we evaluate our LMs on is classification of documents into 6 news categories. The goal of this task is to ascertain how well LMs handle common classification problems. We use a Slovak Categorized News Corpus (Hladek et al., 2014) that contains 4.7K news articles classified into 6 classes: Sports, Politics, Culture, Economy, Health and World. We do not use the *Culture* category, since it contains significantly smaller number of samples.

Unfortunately, no existing work has used this dataset for document classification, so there are no existing results publicly available. We use the same set of baselines and LM fine-tuning as in the case of sentiment analysis, since both these tasks are text classification tasks, see Section 4.4 for more details.

4.5.1 Results

The main results from our experiment are shown in Table 5. We can see that the LMs are again the best performing approach. In this case, the results are quite similar with *SlovakBERT* being the best by a narrow margin. The baselines achieved significantly worse results. Note that our sentence embedding model has the worst results on this task, while it had competitive performance in sentiment classification. We theorize, that the sentence embedding model was trained on sentences and is therefore less capable of handling longer texts, typical for the dataset used here.

Model	F1
TF-IDF	0.953
fastText	0.963
fastText w/ emb.	0.963
STS embeddings	0.935
WikiBERT	0.935
MBERT	0.985
XLM-R-Base	0.987
XLM-R-Large	0.985
Our model	**0.990**

Table 5: Macro-F1 scores for document classification task.

5 Conclusions

We have trained and published *SlovakBERT* – a new large-scale transformers-based Slovak masked language model using 19.35GB of Web-crawled Slovak text. We proposed an evaluation benchmark with multiple tasks for Slovak language and evaluated several models. We conclude, that *SlovakBERT* achieves state-of-the-art results on this benchmark, but multilingual language models are still competitive, especially larger but computationally less efficient models such as XLM-R-Large. We also release the fine-tuned models for the Slovak community.

The lack of evaluation benchmarks is still an issue for many mid-resource language, i.e. languages that have sizeable corpus of text available on the Web, but they do not have annotated natural language understanding datasets available. Our work was limited by this as well, as we were forced to used datasets that created by machine translation (in case of STS), noisy datasets (in case of sentiment analysis) or datasets with almost saturated performance (in case of document classification). Creating new high-quality datasets for the evaluation of Slovak is our future work.

6 Ethical Consideration

SlovakBERT was trained using a Web-crawled corpus. This is a common practice in current NLP, yet, it raises some ethical concerns. Models trained with huge poorly documented corpora might encode in them various societal biases. The Slovak texts written on the Web are not representative of all the Slovak users. Certain demographics groups might be underrepresented and the model might not reflect them accordingly. We do not study these effects in this work and we do not recommend using our model for sensitive applications without further analysis. Unfortunately, there are no datasets, benchmarks or other resources able to measure these effects in Slovak language as of yet.

References

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

Slawomir Dadas, Michal Perelkiewicz, and Rafal Poswiata. 2020. Pre-training polish transformer-based language models at scale. In *Artificial Intelligence and Soft Computing - 19th International Conference, ICAISC 2020, Zakopane, Poland, October 12-14, 2020, Proceedings, Part II*, volume 12416 of *Lecture Notes in Computer Science*, pages 301–314. Springer.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding. *CoRR*, abs/2007.01852.

Daniel Hladek, Jan Stas, and Jozef Juhar. 2014. The Slovak categorized news corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1705–1708, Reykjavik, Iceland. European Language Resources Association (ELRA).

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2485–2494, Hong Kong, China. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland. Association for Computational Linguistics.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Igor Mozetič, Miha Grčar, and Jasmina Smailović. 2016. Multilingual twitter sentiment classification: The role of human annotators. *PloS one*, 11(5).

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90, Online. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Samuel Pecar, Marian Simko, and Maria Bielikova. 2019. Improving sentiment classification in Slovak language. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 114–119, Florence, Italy. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. 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)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Sampo Pyysalo, Jenna Kanerva, Antti Virtanen, and Filip Ginter. 2021. WikiBERT models: Deep transfer learning for many languages. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 1–10, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jakub Sido, Ondrej Prazák, Pavel Pribán, Jan Pasek, Michal Seják, and Miloslav Konopík. 2021. Czert - czech bert-like model for language representation. *CoRR*, abs/2103.13031.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Patrick Xia, Shijie Wu, and Benjamin Van Durme. 2020. Which *BERT? A survey organizing contextualized encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7516–7533, Online. Association for Computational Linguistics.

Daniel Zeman. 2017. Slovak dependency treebank in universal dependencies. *Journal of Linguistics/Jazykovednỳ casopis*, 68(2):385–395.

A Hyperparameter Values

Hyperparameter	Range	Selected
Learning rate	$[10^{-7}, 10^{-3}]$	10^{-5}
Batch size	$\{8, 16, 32, 64, 128\}$	32
Warmup steps	$\{0, 500, 1000, 2000\}$	1000
Weight decay	$[0, 0.1]$	0.05
Label smoothing	$[0, 0.2]$	0
Learning rate scheduler	Various[3]	`linear`

Table 6: Hyperparameters used for POS tagging. Adam was used as an optimization algorithm.

Hyperparameter	Range	Selected
Learning rate	$[10^{-7}, 10^{-3}]$	10^{-5}
Batch size	$\{8, 16, 32, 64, 128\}$	32
Warmup steps	$\{0, 500, 1000, 2000\}$	1000
Weight decay	$[0, 0.2]$	0.15
Learning rate scheduler	Various[4]	`cosine with hard restarts`

Table 7: Hyperparameters used for STS tagging. Adam was used as an optimization algorithm.

[3]See the list of schedulers supported by Hugging Face Transformers library.

[4]See the list of schedulers supported by Sentence Transformers library.

B Tagging Schemata

<table>
<tr><th colspan="2">XPOS</th><th colspan="2">UPOS</th></tr>
<tr><th>Tag</th><th>Description</th><th>Tag</th><th>Description</th></tr>
<tr><td>A</td><td>adjective</td><td rowspan="2">ADJ</td><td rowspan="2">adjective</td></tr>
<tr><td>G</td><td>participle</td></tr>
<tr><td>E</td><td>preposition</td><td>ADP</td><td>adposition</td></tr>
<tr><td>D</td><td>adverb</td><td>ADV</td><td>adverb</td></tr>
<tr><td>Y</td><td>conditional morpheme</td><td rowspan="2">AUX</td><td rowspan="2">auxiliary</td></tr>
<tr><td rowspan="2">V</td><td rowspan="2">verb</td></tr>
<tr><td>VERB</td><td>verb</td></tr>
<tr><td rowspan="2">O</td><td rowspan="2">conjuction</td><td>CCONJ</td><td>coordinating conjunction</td></tr>
<tr><td>SCONJ</td><td>subordinating conjunction</td></tr>
<tr><td rowspan="2">P</td><td rowspan="2">pronoun</td><td>DET</td><td>determiner</td></tr>
<tr><td rowspan="2">PRON</td><td rowspan="2">pronoun</td></tr>
<tr><td>R</td><td>reflexive pronoun</td></tr>
<tr><td>J</td><td>interjection</td><td>INTJ</td><td>interjection</td></tr>
<tr><td rowspan="2">S</td><td rowspan="2">noun</td><td>NOUN</td><td>noun</td></tr>
<tr><td>PROPN</td><td>proper noun</td></tr>
<tr><td>N</td><td>numeral</td><td rowspan="2">NUM</td><td rowspan="2">numeral</td></tr>
<tr><td>0</td><td>digit</td></tr>
<tr><td>T</td><td>particle</td><td>PART</td><td>particle</td></tr>
<tr><td>Z</td><td>punctuation</td><td>PUNCT</td><td>punctuation</td></tr>
<tr><td>W</td><td>abbreviation</td><td rowspan="4">X</td><td rowspan="4">other</td></tr>
<tr><td>Q</td><td>unidentifiable</td></tr>
<tr><td>#</td><td>non-word element</td></tr>
<tr><td>%</td><td>citation in foreign language</td></tr>
<tr><td></td><td></td><td>SYM</td><td>symbol</td></tr>
</table>

Table 8: Slovak POS tagsets and their mapping (Zeman, 2017).

Label	Meaning
0	The two sentences are completely dissimilar.
1	The two sentences are not equivalent, but are on the same topic.
2	The two sentences are not equivalent, but share some details.
3	The two sentences are roughly equivalent, but some important information differs.
4	The two sentences are mostly equivalent, but some unimportant details differ.
5	The two sentences are completely equivalent, as they mean the same thing.

Table 9: Annotation schema for STS datasets (Marelli et al., 2014).