Are the Multilingual Models Better? Improving Czech Sentiment with Transformers

Pavel Přibáň[1,2] and **Josef Steinberger**[1,2]

University of West Bohemia, Faculty of Applied Sciences, Czech Republic
[1]NTIS – New Technologies for the Information Society,
[2]Department of Computer Science and Engineering,
{pribanp,jstein}@kiv.zcu.cz
http://nlp.kiv.zcu.cz

Abstract

In this paper, we aim at improving Czech sentiment with transformer-based models and their multilingual versions. More concretely, we study the task of polarity detection for the Czech language on three sentiment polarity datasets. We fine-tune and perform experiments with five multilingual and three monolingual models. We compare the monolingual and multilingual models' performance, including comparison with the older approach based on recurrent neural networks. Furthermore, we test the multilingual models and their ability to transfer knowledge from English to Czech (and vice versa) with zero-shot cross-lingual classification. Our experiments show that the huge multilingual models can overcome the performance of the monolingual models. They are also able to detect polarity in another language without any training data, with performance not worse than 4.4 % compared to state-of-the-art monolingual trained models. Moreover, we achieved new state-of-the-art results on all three datasets.

1 Introduction

In recent years, BERT-like models (Devlin et al., 2019) based on the Transformer architecture (Vaswani et al., 2017) and generalized language models brought a significant improvement in performance in almost any NLP task (Raffel et al., 2020a), especially in English. Despite this fact, not much work has been recently done in sentiment analysis for the Czech language with the latest Transformer models. We partly fill this gap by focusing on the *Sentiment Classification* task, also known as *Polarity Detection*.

Polarity detection is a classification task where the goal is to assign a sentiment polarity to a given text. The *positive, negative* and *neutral* classes are usually used as the polarity labels. The polarity can also be defined with a different number of labels, i.e., fine-grained sentiment analysis (Liu, 2012).

The models based on BERT were almost exclusively trained for English, limiting their usage to other languages. Recently, however, their cross-lingual adaptions like mBERT (Devlin et al., 2019), mT5 (Xue et al., 2020), XLM (Conneau and Lample, 2019) or XLM-R (Conneau et al., 2020) emerged along with other non-English monolingual versions, for example, Czech (Sido et al., 2021), French (Martin et al., 2020; Le et al., 2019), Arabic (Safaya et al., 2020), Romanian (Dumitrescu et al., 2020), Dutch (Vries et al., 2019) or Finnish (Virtanen et al., 2019).

Our motivation is to reveal the performance limits of the current SotA transformer-based models on the Czech polarity detection task, check the ability of the multilingual models to transfer knowledge between languages and unify the procedure and data that enable the correct future evaluation of this task.

In this paper, we focus on the task of polarity detection applied on Czech text by comparing the performance of seven pre-trained transformer-based models (both monolingual and multilingual) on three Czech datasets. We fine-tune each model on each dataset and we provide a comprehensive survey of their performance. Our experiments show the effectiveness of the Transformer models that significantly outperform the older approaches based on recurrent neural networks. We observe that the monolingual models can be notably outperformed by the multilingual models, but only by those with much more parameters. Moreover, we achieve new state-of-the-art results on all three evaluated datasets.

We are also interested in the ability of the multilingual models to transfer knowledge between languages and its usability for polarity detection. Thus, we perform zero-shot cross-lingual classification,

Proceedings of Recent Advances in Natural Language Processing, pages 1138–1149
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_128

fine-tune four cross-lingual transformer-based models on the English dataset and then test the models on Czech data. We also perform the same experiment in the reverse direction, i.e., from Czech to English. The results reveal that the `XLM-R-Large` model (fine-tuned solely on English) can achieve very competitive results that are only about 4 % worse than the SotA model fine-tuned by us on Czech data. To the best of our knowledge, this is the first paper that performs zero-shot cross-lingual polarity detection for the Czech language.

We also noticed that the comparison with the previous works is rather problematic and thus, we provide a split for all Czech datasets that allows comparing future works much easier. Our code and pre-trained models are publicly available[1].

Our main contributions are the following: 1) We provide the comprehensive performance comparison of the currently available transformer-based models for the Czech language on the polarity detection task along with the models' optimal settings. 2) We test the ability of the multilingual models to transfer knowledge between Czech and English. 3) We release all the fine-tuned models and code freely for research purposes and we provide a data split that allows future comparison and evaluation. Furthermore, we achieved new state-of-the-art results for all three evaluated datasets.

2 Related Work

The previous approaches (Kim, 2014; Johnson and Zhang, 2016; Cliche, 2017; Baziotis et al., 2017; Gray et al., 2017; Conneau et al., 2017) for English polarity detection and other related tasks mostly relied on transfer learning and pre-trained word embeddings such as word2vec (Mikolov et al., 2013) and fastText (Bojanowski et al., 2017) in combinations with Convolutional Neural Networks (CNN) or Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), eventually in conjunction with the modified attention mechanism (Bahdanau et al., 2015; Rocktäschel et al., 2015; Raffel and Ellis, 2015). Furthermore, the new contextualized word representations such as CoVe (McCann et al., 2017) or ELMo (Peters et al., 2018) and pre-trained language model ULMFiT (Howard and Ruder, 2018) were successfully applied to the polarity detection. Finally, the latest transformer-based models like BERT (Devlin et al., 2019), GPT (Radford et al., 2018), RoBERTa (Liu et al., 2019) or T5 (Raffel et al., 2020b) that are all in general trained on language modeling tasks proved their performance superiority for English over all previous approaches, for example in (Sun et al., 2019). These models are pre-trained on a modified language modeling tasks with a huge amount of unlabeled data. In the end, they are fine-tuned for a specific downstream task.

The initial works on Czech polarity detection and sentiment analysis usually used lexical features (Steinberger et al., 2011; Veselovská et al., 2012) or Bag-of-Words text representations along with the Naive Bayes or logistic regression classifiers (Habernal et al., 2013) or a combination of supervised and unsupervised approach (Brychcín and Habernal, 2013). Lenc and Hercig (2016) applied CNN using the architecture from (Kim, 2014) and the LSTM neural network to all three datasets that we use in this paper. Another usage of LSTM neural network with the self-attention mechanism (Humphreys and Sui, 2016) can be found in (Libovickỳ et al., 2018). Similarly, Sido and Konopík (2019) tried to use curriculum learning with CNN and LSTM.

Lehečka et al. (2020) pre-trained a BERT-based model for polarity detection with an improved pooling layer and distillation of knowledge technique. The most recent application of the Transformer model is in (Sido et al., 2021). The authors created a BERT model for Czech and, as one of the evaluation tasks, they performed polarity detection on the FB and CSFD datasets.

To the best of our knowledge, there are no previous works that focus on the zero-shot cross-lingual polarity detection task in the Czech language. The recent related work can be found in (Eriguchi et al., 2018), where the authors use the neural machine translation encoder-based model and English data to perform zero-shot cross-lingual sentiment classification on French. In (Eriguchi et al., 2018) the authors performed the zero-shot classification from Slovene to Croatian. Another related work can be found in (Wang and Banko, 2021; Qin et al., 2020).

3 Data

To the best of our knowledge, there are three Czech publicly available datasets for the polarity detection task: (1) movie review dataset (CSFD), (2) Facebook dataset (FB) and (3) product review dataset (Mallcz), all of them come from (Habernal et al.,

[1] `https://github.com/pauli31/improving-czech-sentiment-transformers`

2013) and each text sample is annotated with one of three[2] labels, i.e., *positive, neutral* and *negative*, see Table 1 for the class distribution. For the cross-lingual experiments we use the two-class English movie review dataset (IMDB) (Maas et al., 2011).

Dataset	Part	Positive	Negative	Neutral	Total
CSFD	train	22 117	21 441	22 235	65 793
	dev	2 456	2 399	2 456	7 311
	test	6 324	5 876	6 077	18 277
	total	30 897	29 716	30 768	91 381
FB	train	1 605	1 227	3 311	6 143
	dev	171	151	361	683
	test	811	613	1 502	2 926
	total	2 587	1 991	5 174	9 752
Mallcz	train	74 100	7 498	23 022	104 620
	dev	8 253	848	2 524	11 625
	test	20 624	2 041	6 397	29 062
	total	102 977	10 387	31 943	145 307
IMDB	train	12 500	12 500	-	25 000
	test	12 500	12 500	-	25 000
	total	25 000	25 000	-	50 000

Table 1: Datasets statistics.

The **FB** dataset contains 10k random posts from nine different Facebook pages that were manually annotated by two annotators. The **CSFD** dataset is created from 90k Czech movie reviews from the Czech movie database[3] that were downloaded and annotated according to their star rating (0–2 stars as *negative*, 3–4 stars as *neutral*, 5–6 stars as *positive*). The **Mallcz** dataset consists of 145k users' reviews of products from Czech e-shop[4], the labels are assigned according to the review star rating on the scale 0-5, where the reviews with 0-3 stars are labeled as *negative*, 4 stars as *neutral* and 5 stars as *positive*. The English **IMDB** dataset includes 50k movie reviews scraped from the Internet Movie Database[5] with *positive* and *negative* classes split into training and testing parts of equal size.

Since there is no official partitioning for the Czech datasets, we split them into training, development and testing parts with the same class distribution for each part as it is in the original dataset, see Table 1. For the Mallcz and CSFD datasets, we use the following ratio: 80 % for training data, 20 % for testing data, for the FB dataset, it is 70 % and 30 %, respectively and 10 % from the training data (for all datasets) is used as the development data. We used different split ratio for the FB dataset because it is approximately ten and sixteen times smaller than the CSFD and Mallcz datasets, respectively and we did not want to reduce the size of the testing data too much.

4 Models Description

We performed exhaustive experiments with transformed-based models and in order to compare them with the previous works, we also implemented the older models (baseline models) that include the logistic regression classifier and the BiLSTM neural network.

4.1 Baseline Models

We re-implemented the best models from (Habernal et al., 2013), i.e., logistic regression classifier (**lrc**) with character n-grams (in a range from 3-grams to 6-grams), word uni-grams and bi-grams features. The second baseline model is the **LSTM** model partly inspired by (Baziotis et al., 2017). Its input is a sequence of t tokens represented as a matrix $\mathbf{M} \in \mathbb{R}^{t \times d}$, where $d = 300$ is a dimension of the Czech pre-trained fastText word embeddings (Bojanowski et al., 2017)[6]. The maximal size of the input vocabulary is set to 300 000. The input is passed into the trainable embedding layer that is followed by two BiLSTM (Graves and Schmidhuber, 2005) layers and after each, the dropout (Srivastava et al., 2014) is applied. After the two BiLSTM layers, the self-attention mechanism is applied. The output is then passed to a fully-connected softmax layer. An output of the softmax layer is a probability distribution over the possible classes. We use the Adam (Kingma and Ba, 2014) optimizer with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) and with weight decay modification (Loshchilov and Hutter, 2017) and the cross-entropy loss function. We replace numbers, emails and links with generic tokens, we tokenize input text with the TokTok tokenizer[7] and we use a customized stemmer[8].

We use different hyper-parameters for each dataset, see Appendix A.1 for the complete settings.

[2]The FB dataset also contains 248 samples with a fourth class called *bipolar*, but we ignore this one.

[3]https://www.csfd.cz

[4]https://www.mall.cz

[5]https://www.imdb.com

[6]Available at https://fasttext.cc/docs/en/crawl-vectors.html

[7]https://github.com/jonsafari/tok-tok

[8]https://github.com/UFAL-DSG/alex/blob/master/alex/utils/czech_stemmer.py

4.2 Transformer Models

In total, we use eight different transformer-based models (five of them are multilingual). All of them are based on the original BERT model. They use only the encoder part of the original Transformer (Vaswani et al., 2017), although their pre-training procedure may differ. There are also *text-to-text* models like T5 (Raffel et al., 2020b) and BART (Lewis et al., 2019) and their multilingual versions mT5 (Xue et al., 2020) and mBART (Liu et al., 2020; Tang et al., 2020). The main difference from BERT-like models is that they use the full encoder-decoder architecture of the Transformer. They are mainly intended for text generation tasks (e.g., abstractive summarization). We decided to use only the BERT-like models with the same architecture because they fit more for the classification task.

All the models are pre-trained on a modified language modeling task, for example, *Masked Language Modeling* (MLM) and eventually on some classification task like *Next Sentence Prediction* (NSP) or *Sentence Ordering Prediction* (SOP), see (Devlin et al., 2019; Lan et al., 2020) for details. The evaluated models differ in the number of parameters (see Table 2) and thus, their performance is also very different, see Section 5.

Model	#Params	Vocab	#Langs
Czert-B	110M	30k	1
Czert-A	12M	30k	1
RandomALBERT	12M	30k	1
mBERT	177M	120k	104
SlavicBERT	177M	120k	4
XLM	570M	200k	100
XLM-R-Base	270M	250k	100
XLM-R-Large	559M	250k	100

Table 2: Models statistics with a number of parameters, vocabulary size and a number of supported languages.

Czert-B is Czech version of the of the original $\text{BERT}_{\text{BASE}}$ model (Devlin et al., 2019). The only difference is that during the pre-training, the authors increased the batch size to 2048 and they slightly modified the NSP prediction task (Sido et al., 2021).

Czert-A is the Czech version of the ALBERT model (Lan et al., 2020), also with the same modification as `Czert-B`, i.e., batch size was set to 2048 and the modified NSP prediction task is used instead of the SOP task (Sido et al., 2021).

RandomALBERT we follow the evaluation in (Sido et al., 2021) and we also test randomly initialized ALBERT model without any pre-training to show the importance of pre-training of such models and its performance influence on the polarity detection task.

mBERT (Devlin et al., 2019) is a multilingual version of the original $\text{BERT}_{\text{BASE}}$, jointly trained on 104 languages.

SlavicBERT (Arkhipov et al., 2019) is initialized from the `mBERT` checkpoint and further pre-trained with a modified vocabulary only for four Slavic languages (Bulgarian, Czech, Polish and Russian).

XLM (Conneau and Lample, 2019) utilizes the training procedure of the original BERT model for multilingual settings mainly by using the Byte-Pair Encoding (BPE) and increasing the shared vocabulary between languages.

XLM-R-Base (Conneau et al., 2020) is a multilingual version of the RoBERTa (Liu et al., 2019) specifically optimized and pre-trained for 100 languages.

XLM-R-Large (Conneau et al., 2020) is the same model as the `XLM-R-Base`, but it is larger (it has more parameters).

4.3 Transformers Fine-Tuning

To utilize the models for text classification, we follow the default approaches mentioned in the corresponding models' papers and we fine-tune all parameters of the models. In all models except `XLM`, we use the final hidden vector $\mathbf{h} \in \mathbb{R}^H$ of the special classification token `[CLS]` or `<s>` taken from the pooling layer[9] of BERT or RoBERTa models, respectively. The vector $\mathbf{h}$ represents the entire encoded sequence input, where H is the hidden size of the corresponding model. We add a task-specific linear layer (with a dropout set to 0.1) represented by a matrix $\mathbf{W} \in \mathbb{R}^{|C| \times H}$, where C is a set of classes. We compute the classification output, i.e., the input sample being classified as class $c \in C$ as $c = \text{argmax}\,(\mathbf{h}\mathbf{W}^T)$.

In the case of the `XLM` model, we take the last hidden state (without any pooling layer) of the first input token and we apply the same linear layer ($\mathbf{W} \in \mathbb{R}^{|C| \times H}$) and approach to obtain the classification output. For learning, we use the Adam optimizer with default parameters and with weight decay (same as for the `LSTM` model), and the cross-

[9]The pooling layer is a fully-connected layer of size H with a hyperbolic tangent used as the activation function.

entropy loss function. See Section 5.1 and Appendix A.2 for the hyper-parameters we used.

5 Experiments & Results

We perform two types of experiments, i.e., *monolingual* and *cross-lingual*. In *monolingual* experiments, we fine-tune and evaluate the Transformer models for each dataset separately on three-class (*positive, negative* and *neutral*) and two-class (*positive* and *negative*) sentiment analysis. We also implemented the logistic regression (`lrc`) and `LSTM` baseline models and we compare the results with the existing works.

In *cross-lingual* experiments, we test the ability of four multilingual transformer-based models to transfer knowledge between English and Czech. We run the multilingual models only on the two-class datasets (*positive* and *negative*). We fine-tune either on English (IMDB) or Czech (CSFD), and then we evaluate on the other language. Thus we perform the *zero-shot cross-lingual* classification. We decided to use the IMDB and CSFD dataset because they are from the same domain i.e., movie reviews.

Each experiment[10] was performed at least five times and we report the results using the macro F_1 score.

5.1 Monolingual Experiments

The goal of the monolingual experiments is to reveal the current state-of-the-art performance on the Czech polarity datasets, namely CSFD, FB and Mallcz (see Section 3) and provide a comparison between the available models and their settings.

As we already mentioned, we split the datasets into training, development and testing parts. There is no official split for the datasets and we found out that all the available works usually use either 10-fold cross-validation or they split[11] the datasets on their own, the † and * symbols in Table 3, respectively causing the comparison to be difficult.

We fine-tune all models on training data and we measure the results on the development data. We select the model with the best performance on the development data and we fine-tune the model on combined training and development data. We report the results in Table 3 on the testing data with 95% confidence intervals.

Firstly, we re-implemented the logistic regression classifier (`lrc`) with the best feature combination from (Habernal et al., 2013) and we report the results on our data split. We can see that we obtained very similar results to the ones stated in (Habernal et al., 2013). We also tried to improve this baseline with Tf-idf weighting, but it did not lead to any significant improvements, so we decided to keep the settings the same as in (Habernal et al., 2013), so the results are comparable.

For the `LSTM` model, we tried different combinations of hyper-parameters (learning rate, optimizer, dropout, etc.). We report the used hyper-parameters for the results from Table 3 in Appendix A.2. Our implementation is only about 1 % worse than LSTM with the self-attention model from (Libovický et al., 2018), but they used a different data split. For the Mallcz dataset, we were not able to outperform the `lrc` baseline with the `LSTM` model.

We fine-tune all parameters of the seven pre-trained BERT-based models and one randomly initialized ALBERT model. In our experiments, we use constant learning rate and also linear learning rate decay (without learning rate warm-up) with the following initial learning rates: 2e-6, 2e-5 and 2.5e-5. We got inspired by the ones used in (Sun et al., 2019). Based on the average number of tokens for each dataset and models' tokenizer (see Table 4 and Figures 1, 2, 3)[12], we use a max sequence length of 64 and a batch size of 32 for the FB dataset. We restrict the max sequence length for the CSFD and Mallcz datasets to 512 and use a batch size of 32. All other hyper-parameters of the models are set to the pre-trained models' defaults. See Table 7 in Appendix A.2 for the reported results' hyper-parameters.

We repeated all the basic experiments with the polarity detection task from (Sido et al., 2021) with the new data split. Our results do not significantly differ, as shown in Table 8 and in Appendix A.2. If we compare the BERT model from (Lehečka et al., 2020) with the `Czert-B, mBERT` and `SlavicBERT` models[13], we can see that on the binary task, they also perform very similarly, i.e., around 93 %, but again they used different test data (the entire CSFD dataset[14]). The obvious observation is that the `XLM-R-Large` model is supe-

[10] Except for the experiments with the `lrc` model.

[11] The authors do not provide any recipe to reproduce the results.

[12] The distributions of the other models were similar to those shown in the mentioned Figures.

[13] All of them should have the same or almost the same architecture and a similar number of parameters.

[14] The examples with positive and negative classes.

Model	3 Classes			2 Classes		
	CSFD	FB	Mallcz	CSFD	FB	Mallcz
lrc (ours)	79.63	67.86	76.71	91.42	88.12	88.98
LSTM (ours)	79.88 ± 0.18	72.89 ± 0.49	73.43 ± 0.12	91.82 ± 0.09	90.13 ± 0.17	88.02 ± 0.24
Czert-A	79.89 ± 0.60	73.06 ± 0.59	76.79 ± 0.38	91.84 ± 0.84	91.28 ± 0.18	91.20 ± 0.26
Czert-B	84.80 ± 0.10	76.90 ± 0.38	79.35 ± 0.24	94.42 ± 0.15	93.97 ± 0.30	92.87 ± 0.15
mBERT	82.74 ± 0.16	71.61 ± 0.13	70.79 ± 5.74	93.11 ± 0.29	88.76 ± 0.42	72.79 ± 3.09
SlavicBERT	82.59 ± 0.12	73.93 ± 0.53	75.34 ± 2.54	93.47 ± 0.33	89.84 ± 0.43	90.99 ± 0.15
RandomALBERT	75.79 ± 0.18	62.53 ± 0.46	64.81 ± 0.25	89.99 ± 0.21	81.71 ± 0.56	85.38 ± 0.10
XLM-R-Base	84.82 ± 0.10	77.81 ± 0.50	75.43 ± 0.07	94.32 ± 0.34	93.26 ± 0.74	92.56 ± 0.07
XLM-R-Large	**87.08 ± 0.11**	**81.70 ± 0.64**	**79.81 ± 0.21**	**96.00 ± 0.02**	**96.05 ± 0.01**	**94.37 ± 0.02**
XLM	83.67 ± 0.11	71.46 ± 1.58	77.56 ± 0.08	93.86 ± 0.18	89.94 ± 0.27	91.97 ± 0.22
(Habernal et al., 2013)†	79.00	69.00	75.00	-	90.00	-
(Brychcín and Habernal, 2013)†	81.53 ± 0.30	-	-	-	-	-
(Libovický et al., 2018)*	80.80 ± 0.10	-	-	-	-	-
(Lehečka et al., 2020)*	-	-	-	93.80	-	-

Table 3: The final monolingual results as macro F_1 score for all three Czech polarity datasets on two and three classes. For experiments with neural networks performed by us, we present the results with a 95% confidence interval. The models from papers marked with † were evaluated with 10-fold cross-validation and the ones marked with * were evaluated on custom data split.

Model	CSFD		FB		Mallcz	
	Avg.	Max.	Avg.	Max.	Avg.	Max.
Czert-B	84.5	1000	20.3	64	34.3	1471
mBERT	111.6	1206	25.6	66	46.6	2038
SlavicBERT	83.6	983	20.7	62	34.3	1412
XLM	100.5	1058	22.6	64	41.0	1812
Czert-A RandomALBERT	81.7	993	19.7	62	32.6	1435
XLM-R-Base XLM-R-Large	93.9	952	20.4	53	37.5	1670

Table 4: The average and maximum number of subword tokens for each model's tokenizer and dataset.

rior to all others by a significant margin for any dataset. Only for the three-class Mallcz dataset, the `Czert-B` model is competitive (the confidence intervals almost overlap). From the results for the `RandomALBERT` model, we can see how important is the pre-training phase for Transformers, since the model is even worse than the logistic regression classifier[15].

5.2 Cross-lingual Experiments

The cross-lingual experiments were performed with the multilingual models that support English and Czech. For these experiments, we use linear learning rate decay with an initial learning rate of 2e-6.

Firstly, we fine-tuned the models on the English IMDB dataset and we evaluated them on the test part of the Czech binary CSFD dataset (i.e., zero-shot cross-lingual classification). We randomly selected 5k examples from the IMDB dataset as the development data. The rest of the 45k examples is used as training data. We select the models that perform best on the English development data[16] and we report the results in Table 5. The *test (cs)* column refers to results obtained on the CSFD testing part. For easier comparison, we also include the *Monoling. (cs)* column that contains the results (same as in Table 3) for models trained on Czech data. The `XLM-R-Large` was able to achieve results only about 4.4 % worse than the same model that was fine-tuned on Czech data. It is a great result if we consider that the model has never seen any labeled Czech data. The `XLM` and `mBERT` models perform much worse.

Model	EN → CS		Monoling. (cs)
	dev (en)	test (cs)	
XLM-R-Base	94.52 ± 0.12	88.01 ± 0.28	94.32 ± 0.34
XLM-R-Large	95.86 ± 0.06	**91.61 ± 0.06**	96.00 ± 0.02
XLM	92.76 ± 0.34	75.37 ± 0.29	93.86 ± 0.18
mBERT	93.07 ± 0.03	76.32 ± 1.13	93.11 ± 0.29

Table 5: Macro F_1 score for cross-lingual experiments from English to Czech.

The second type of cross-lingual experiment was performed in a reverse direction, i.e., from Czech to English. We use the Czech CSFD training and testing data for fine-tuning and we evaluate the model on the English IMDB test data. We report the results in Table 6 using the accuracy because

[15]The model was trained for a maximum of 15 epochs and it would probably get better with a higher number of epochs, but the other models were trained for the same or lower number of epochs.

[16]The *dev (en)* column in Table 5.

the current state-of-the-art works (Thongtan and Phienthrakul, 2019; Sun et al., 2019) use this metric. Similarly to the previous case, we selected the model that performs best on Czech CSFD development data. For these experiments, the `mBERT` did not converge. As in the previous experiment, the `XLM-R-Large` performs best and it achieves almost 94 % accuracy that is only 3.4 % below the current SotA result from (Thongtan and Phienthrakul, 2019).

Model	CS → EN	
	dev (cs)	test (en)
XLM-R-Base	94.22 ± 0.01	89.53 ± 0.15
XLM-R-Large	95.65 ± 0.17	**93.98 ± 0.10**
XLM	93.66 ± 0.13	78.24 ± 0.46
(Thongtan and Phienthrakul, 2019)	-	97.42
(Sun et al., 2019)	-	95.79

Table 6: Accuracy results for cross-lingual experiments from Czech to English.

Based on the results, we can conclude that the `XLM-R-Large` model is very capable of transferring knowledge between English and Czech (and probably between other languages as well). It is also important to note that Czech and English are languages from a different language family with a high number of differences both in syntax and grammar.

5.3 Discussion & Remarks

We can see from the results that the recent pre-trained transformer-based models beat the older approaches (`lrc` and `LSTM`) by a large margin. The monolingual `Czert-B` model is in general outperformed only by the `XLM-R-Large` and `XLM-R-Base` models, but these models have five times/three times more parameters, and eight times larger vocabulary. Taking into account these facts, the `Czert-B` model is still very competitive. It may be beneficial in certain situations to use a smaller model like this that does not need such computational resources as the ones that are required by the `XLM-R-Large`.

During the fine-tuning, we observed that in most cases, the lower learning rate 2e-6 (see Table 7 in Appendix A.2) leads to better results. Thus we recommend using the same one or similar order. The higher learning rates tend to provide worse results and the model does not converge.

According to the generally higher confidence interval, the fine-tuning of a smaller dataset like FB that has only about 6k training examples is, in general, less stable and more prone to overfitting than training a model on datasets with tens of thousands of examples. We also noticed that fine-tuning of the `mBERT` and `SlavicBERT` on the Mallcz dataset is very unstable (see the confidence interval in Table 3). Unfortunately, we did not find out the reason. A more detailed error analysis could reveal the reason.

6 Conclusion

In this work, we evaluated the performance of available transformer-based models for the Czech language on the task of polarity detection. We compared the performance of the monolingual and multilingual models and we showed that the large `XLM-R-Large` model can outperform the monolingual `Czert-B` model. The older approach based on recurrent neural networks is surpassed by a very large margin by the Transformers. Moreover, we achieved new state-of-the-art results on all three Czech polarity detection datasets.

We performed zero-shot cross-lingual polarity detection from English to Czech (and vice versa) with four multilingual models. We showed that the `XLM-R-Large` is able to detect polarity in another language without any labeled data. The model performs no worse than 4.4 % in comparison to our new state-of-the-art monolingual model. To the best of our knowledge, this is the first work that aims at cross-lingual polarity detection in Czech. Our code and pre-trained models are publicly available.

In the future work, we intend to perform a deep error analysis to find in which cases the current models fail and compare approaches that use the linear cross-lingual transformations (Artetxe et al., 2018; Brychcín, 2020) that explicitly map semantic spaces into one shared space. The second step in the cross-lingual settings is to employ more than two languages and utilize the models for different domains.

Acknowledgments

This work has been partly supported by ERDF "Research and Development of Intelligent Components of Advanced Technologies for the Pilsen Metropolitan Area (InteCom)" (no.: CZ.02.1.01/0.0/0.0/17 048/0007267); and by Grant No. SGS-2019-018 Processing of heterogeneous data and its specialized applications. Computational resources were supplied by the project "e-Infrastruktura CZ" (e-

INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures.

References

Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. 2019. Tuning multilingual transformers for language-specific named entity recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 89–93, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. DataStories at SemEval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

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.

Tomáš Brychcín. 2020. Linear transformations for cross-lingual semantic textual similarity. *Knowledge-Based Systems*, 187:104819.

Tomáš Brychcín and Ivan Habernal. 2013. Unsupervised improving of sentiment analysis using global target context. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 122–128, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Mathieu Cliche. 2017. BB_twtr at SemEval-2017 task 4: Twitter sentiment analysis with CNNs and LSTMs. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 573–580, Vancouver, Canada. Association for Computational Linguistics.

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*, volume 32. Curran Associates, Inc.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116, Valencia, Spain. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Stefan Dumitrescu, Andrei-Marius Avram, and Sampo Pyysalo. 2020. The birth of Romanian BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4324–4328, Online. Association for Computational Linguistics.

Akiko Eriguchi, Melvin Johnson, Orhan Firat, Hideto Kazawa, and Wolfgang Macherey. 2018. Zero-shot cross-lingual classification using multilingual neural machine translation. *arXiv preprint arXiv:1809.04686*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Scott Gray, Alec Radford, and Diederik P Kingma. 2017. Gpu kernels for block-sparse weights. *arXiv preprint arXiv:1711.09224*, 3.

Ivan Habernal, Tomáš Ptáček, and Josef Steinberger. 2013. Sentiment analysis in Czech social media using supervised machine learning. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 65–74, Atlanta, Georgia. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Glyn W Humphreys and Jie Sui. 2016. Attentional control and the self: the self-attention network (san). *Cognitive neuroscience*, 7(1-4):5–17.

Rie Johnson and Tong Zhang. 2016. Supervised and semi-supervised text categorization using lstm for region embeddings. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 526–534. JMLR.org.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2019. Flaubert: Unsupervised language model pre-training for french. *arXiv preprint arXiv:1912.05372*.

Jan Lehečka, Jan Švec, Pavel Ircing, and Luboš Šmídl. 2020. Bert-based sentiment analysis using distillation. In *Statistical Language and Speech Processing*, pages 58–70, Cham. Springer International Publishing.

Ladislav Lenc and Tomás Hercig. 2016. Neural networks for sentiment analysis in czech. In *Proceedings of the 16th ITAT: Slovenskočeský NLP workshop (SloNLP 2016)*, volume 1649 of *CEUR Workshop Proceedings*, pages 48–55, Bratislava, Slovakia. Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, CreateSpace Independent Publishing Platform.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jindrich Libovickỳ, Rudolf Rosa, Jindrich Helcl, and Martin Popel. 2018. Solving three czech nlp tasks with end-to-end neural models. In *ITAT*, pages 138–143.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

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.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. 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.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Matthew 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.

Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP. *CoRR*, abs/2006.06402.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Colin Raffel and Daniel PW Ellis. 2015. Feed-forward networks with attention can solve some long-term memory problems. *arXiv preprint arXiv:1512.08756*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020a. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020b. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media.

Jakub Sido and Miloslav Konopík. 2019. Curriculum learning in sentiment analysis. In *Speech and Computer*, pages 444–450, Cham. Springer International Publishing.

Jakub Sido, Ondřej Pražák, Pavel Přibáň, Jan Pašek, Michal Seják, and Miloslav Konopík. 2021. Czert–czech bert-like model for language representation. *arXiv preprint arXiv:2103.13031*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Josef Steinberger, Polina Lenkova, Mohamed Ebrahim, Maud Ehrmann, Ali Hurriyetoglu, Mijail Kabadjov, Ralf Steinberger, Hristo Tanev, Vanni Zavarella, and Silvia Vázquez. 2011. Creating sentiment dictionaries via triangulation. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 28–36, Portland, Oregon. Association for Computational Linguistics.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Tan Thongtan and Tanasanee Phienthrakul. 2019. Sentiment classification using document embeddings trained with cosine similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 407–414, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Katerina Veselovská, Jan Hajic, and Jana Sindlerová. 2012. Creating annotated resources for polarity classification in czech. In *KONVENS*, pages 296–304.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. *arXiv:1912.09582 [cs]*.

Cindy Wang and Michele Banko. 2021. Practical transformer-based multilingual text classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 121–129, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mT5: A massively multilingual pre-trained text-to-text transformer.

A Appendix

A.1 LSTM Hyper-parameters

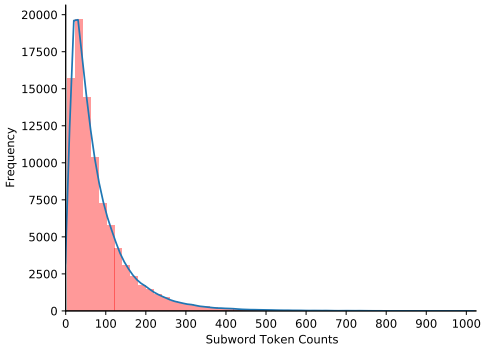
We use cross-entropy as the loss function and the Adam (Kingma and Ba, 2014) optimizer with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) and with a modification from (Loshchilov and Hutter, 2017) for the FB dataset. The embedding layer is trainable with a maximum size of 300k. The max sequence length for the input t tokens is 64 for the FB dataset and 512 for the CSFD and Mallcz dataset with weight decay in the optimizer set to 0. We use Czech pre-trained fastText (Bojanowski et al., 2017) embeddings[17]. For the Mallcz and CSFD datasets, we use 128 units in the BiLSTM layers and a batch size of 128. For the FB dataset, we use 256 units in the BiLSTM layers and a batch size of 256 with weight decay in the optimizer set to 1e-4.

For all datasets, we use 10 % of total steps (batch updates) to warm up the learning rate, which means that during the training, the linear rate is firstly linearly increasing to the initial learning rate before being decayed with the corresponding learning rate scheduler. The dropout after the BiLSTM layers is set to 0.2. We use cosine (the * symbol in Table 7) and the exponential learning rate scheduler (the ‡ symbol in Table 7) with a decay rate set to 0.05. Table 7 contains the initial learning rate and the number of epochs for the `LSTM` model for each dataset.

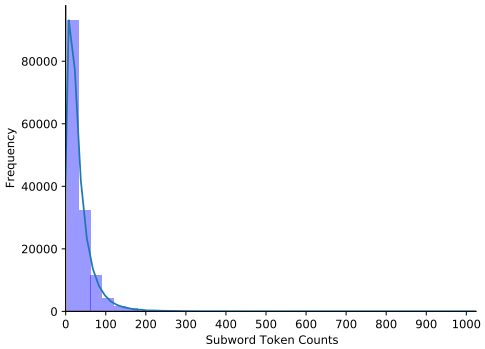
A.2 Transformer Hyper-parameters

For fine-tuning of the transformer-based models, we use the same modification (Loshchilov and Hutter, 2017) of the Adam (Kingma and Ba, 2014) optimizer with default weight decay set to 1e-2. We use different learning rates and a number of epochs for each combination of the models and datasets, see Table 7. We use either constant linear rate or linear learning rate decay without learning rate warm-up. We use default values of all other hyper-parameters.

For the cross-lingual experiments, we use only the linear learning rate decay scheduler with the initial learning rate set to 2e-6 without learning rate warm-up. For the cross-lingual experiments from English to Czech, the numbers of epochs used for the fine-tuning are 5, 2, 4 and 10 for the `XLM-R-Base`, `XLM-R-Large`, `XLM` and `mBERT` models, respectively. For the cross-lingual experiments from Czech to English, the numbers of epochs used for the fine-tuning are 25, 5 and 9 for the `XLM-R-Base`, `XLM-R-Large` and `XLM` models[18], respectively.



(a) CSFD – Czert-B



(b) Mallcz – Czert-B

Figure 1: Subword token histograms for the CSFD and Mallcz datasets for the `Czert-B` model.

A.3 Computational Cluster

For fine-tuning of the Transformers models we use the Czech national cluster Metacentrum[19]. We use two NVIDIA A100 GPUs each with 40GB memory.

[17] Available at `https://fasttext.cc/docs/en/crawl-vectors.html`

[18] The `mBERT` model did not converge for this experiment

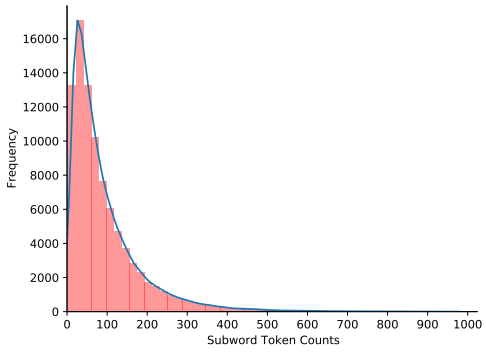
[19] See `https://wiki.metacentrum.cz/wiki/Usage_rules/Acknowledgement`

Model	3 Classes			2 Classes		
	CSFD	FB	Mallcz	CSFD	FB	Mallcz
Log. reg. (ours)	79.63	67.86	76.71	91.42	88.12	88.98
LSTM (ours)	79.88 ± 0.18 (5e-4 / 2)*	72.89 ± 0.49 (5e-4 / 5)*	73.43 ± 0.12 (5e-4 / 10) ‡	91.82 ± 0.09 (5e-4 / 2)*	90.13 ± 0.17 (5e-4 / 5)*	88.02 ± 0.24 (5e-4 / 2)‡
Czert-A	79.89 ± 0.60 (2e-6 / 8)	73.06 ± 0.59 (2e-5 / 8)	76.79 ± 0.38 (2e-5 / 12)	91.84 ± 0.84 (2e-5 / 8)	91.28 ± 0.18 (2e-5 / 15)†	91.20 ± 0.26 (2e-5 / 14)
Czert-B	84.80 ± 0.10 (2e-5 / 12)	76.90 ± 0.38 (2e-6 / 5)†	79.35 ± 0.24 (2e-5 / 15)	94.42 ± 0.15 (2e-5 / 15)	93.97 ± 0.30 (2e-5 / 2)	92.87 ± 0.15 (2e-5 / 15)
mBERT	82.74 ± 0.16 (2e-6 / 13)	71.61 ± 0.13 (2e-6 / 13)†	70.79 ± 5.74 (2e-5 / 10)	93.11 ± 0.29 (2e-6 / 14)†	88.76 ± 0.42 (2e-5 / 8)	72.79 ± 3.09 (2e-5 / 1)
SlavicBERT	82.59 ± 0.12 (2e-6 / 12)	73.93 ± 0.53 (2e-5 / 4)	75.34 ± 2.54 (2e-5 / 10)	93.47 ± 0.33 (2e-6 / 15)†	89.84 ± 0.43 (2e-5 / 9)†	90.99 ± 0.15 (2e-6 / 14)†
RandomALBERT	75.79 ± 0.18 (2e-6 / 14)	62.53 ± 0.46 (2e-6 / 14)†	64.81 ± 0.25 (2e-6 / 15)†	89.99 ± 0.21 (2e-6 / 14)†	81.71 ± 0.56 (2e-6 / 15)†	85.38 ± 0.10 (2e-6 / 14)†
XLM-R-Base	84.82 ± 0.10 (2e-6 / 15)†	77.81 ± 0.50 (2e-6 / 7)†	75.43 ± 0.07 (2e-6 / 15)†	94.32 ± 0.34 (2e-6 /14) †	93.26 ± 0.74 (2e-6 / 5)†	92.56 ± 0.07 (2e-6 / 12)†
XLM-R-Large	**87.08 ± 0.11** (2e-6 / 11)	**81.70 ± 0.64** (2e-6 / 5)†	**79.81 ± 0.21** (2e-6 / 24)†	**96.00 ± 0.02** (2e-6 / 143)†	**96.05 ± 0.01** (2e-6 / 15)	**94.37 ± 0.02** (2e-6 / 15)†
XLM	83.67 ± 0.11 (2e-5 / 11)	71.46 ± 1.58 (2e-6 / 9)†	77.56 ± 0.08 (2e-6 / 14)†	93.86 ± 0.18 (2e-5 / 5)	89.94 ± 0.27 (2e-6 / 15)†	91.97 ± 0.22 (2e-6 / 16)†

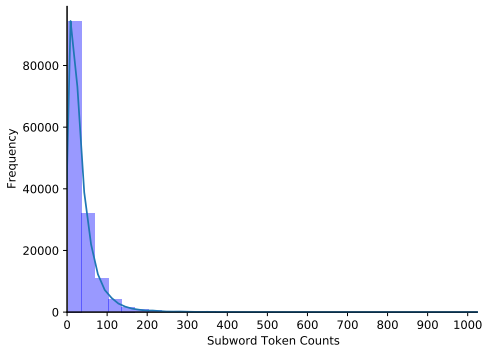
Table 7: The final monolingual results as macro F_1 score and hyper-parameters for all three Czech polarity datasets on two and three classes. For experiments with neural networks performed by us, we present the results with a 95% confidence interval. For each result, we state the used learning rate and the number of epochs used for the training. The † symbol denotes that the result was obtained with constant learning rate, $*$ denotes the cosine learning rate decay, ‡ denotes exponential learning rate decay, otherwise the linear learning rate decay was used.

Model	CSFD		FB	
	(Sido et al., 2021)	Ours	(Sido et al., 2021)	Ours
mBERT	82.80 ± 0.14 (2e-6 / 13)	82.74 ± 0.16 (2e-6 / 13)	71.72 ± 0.91 (2e-5 / 6)	71.61 ± 0.13 (2e-6 / 13)
SlavicBERT	82.51 ± 0.14 (2e-6 / 12)	82.59 ± 0.12 (2e-6 / 12)	73.87 ± 0.50 (2e-5 / 3)	73.93 ± 0.53 (2e-5 / 4)
RandomALBERT	75.40 ± 0.18 (2e-6 / 13)	75.79 ± 0.18 (2e-6 / 14)	59.50 ± 0.47 (2e-6 / 14)	62.53 ± 0.46 (2e-6 / 14)
Czert-A	79.58 ± 0.46 (2e-6 / 8)	79.89 ± 0.60 (2e-6 / 8)	72.47 ± 0.72 (2e-5 / 8)	73.06 ± 0.59 (2e-5 / 8)
Czert-B	84.79 ± 0.26 (2e-5 / 12)	84.80 ± 0.10 (2e-5 / 12)	76.55 ± 0.14 (2e-6 / 12)	76.90 ± 0.38 (2e-6 / 5)

Table 8: Comparison of results from (Sido et al., 2021) with results obtained by us.

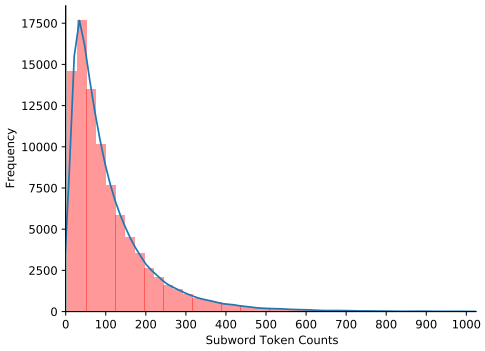


(a) CSFD – XLM-R-Base and XLM-R-Large

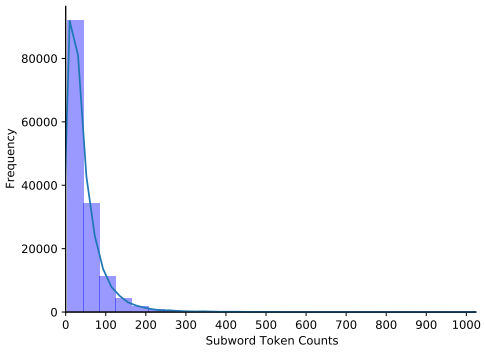


(b) Mallcz – XLM-R-Base and XLM-R-Large

Figure 2: Subword token histograms for the CSFD and Mallcz datasets for the `XLM-R-Base` and `XLM-R-Large` models.



(a) CSFD – mBERT



(b) Mallcz – mBERT

Figure 3: Subword token histograms for the CSFD and Mallcz datasets for the `mBERT` model.