Czert – Czech BERT-like Model for Language Representation

Jakub Sido* and **Ondřej Pražák*** and **Pavel Přibáň**
and **Jan Pašek** and **Michal Seják** and **Miloslav Konopík**

{sidoj,ondfa,pribanp,pasekj,sejakm,konopik}@kiv.zcu.cz

NTIS – New Technologies for the Information Society,
Department of Computer Science and Engineering,
Faculty of Applied Sciences, University of West Bohemia, Technická 8, 306 14 Plzeň
Czech Republic

Abstract

This paper describes the training process of the first Czech monolingual language representation models based on BERT and ALBERT architectures. We pre-train our models on more than 340K of sentences, which is 50 times more than multilingual models that include Czech data. We outperform the multilingual models on 9 out of 11 datasets. In addition, we establish the new state-of-the-art results on nine datasets. At the end, we discuss properties of monolingual and multilingual models based upon our results. We publish all the pre-trained and fine-tuned models freely for the research community.

1 Introduction

Transfer learning and pre-trained word embeddings became a crucial component for most Natural Language Processing (NLP) models. Contextualized methods (McCann et al., 2017; Peters et al., 2018; Howard and Ruder, 2018) overcame the initial context insensitive word embeddings approaches (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017). (McCann et al., 2017; Peters et al., 2018). The word representations generated by the named methods are usually used as input features for other task-specific models that are further trained. Starting with the BERT (Devlin et al., 2018), the BERT-like models (Lan et al., 2020; Liu et al.; Sanh et al., 2019; Yang et al., 2019) based on Transformer architecture (Vaswani et al., 2017), achieved a significant performance improvement in many NLP tasks (Raffel et al., 2019). These recent models are trained on a language model task or tasks that are closely related to it. Such pre-training allows them to capture the general representation of language and text. The pre-trained models are then directly fine-tuned with specific data for a selected downstream task. The performance improvement of these models is paid by the vastly increased requirements (i.e., data and computational resources) for their training.

The mentioned models are primarily trained for English. Recently, models for other, mostly larger, languages have been released, e.g., French (Martin et al., 2020; Le et al., 2019), Polish (Kłeczek, 2020), Turkish (Schweter, 2020), Russian[1], Italian[2], German[2], Arabic (Safaya et al., 2020), but also for languages that are spoken by a relatively small number of people, i.e., Romanian (Dumitrescu et al., 2020), Dutch (Vries et al., 2019) or Finish (Virtanen et al., 2019). There were also introduced multilingual models (Conneau and Lample, 2019; Conneau et al., 2020), that can be used for multiple languages at once but usually at the cost of lower performance in comparison to solely monolingual models (Martin et al., 2020; Virtanen et al., 2019; Dumitrescu et al., 2020) as we show in this paper.

Our main motivation is to train and provide publicly available models[3] for the Czech language that performs better than available multilingual models.

In this paper, we describe a process of training of two BERT-like models for Czech language and their evaluation on six tasks along with a comparison to two multilingual models, i.e. mBERT (Devlin et al., 2018) and SlavicBERT (Arkhipov et al., 2019). More concretely, the architectures of our models are based on the ALBERT (Lan et al., 2020) model (Czert-A) and the original BERT (Devlin et al., 2018) model (Czert-B). Both of our models are trained on a text corpus of the approximate size of 36 GB of plain text consisting of Czech Wikipedia articles, crawled Czech news and Czech

*Equal contribution.

[1] http://docs.deeppavlov.ai/en/master/features/models/bert.html

[2] https://github.com/dbmdz/berts

[3] The model is available at https://github.com/kiv-air/Czert

Proceedings of Recent Advances in Natural Language Processing, pages 1326–1338
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_149

National Corpus (Křen et al., 2016). We train the models from scratch (i.e., with random initialization) using *Masked Language Model* (MLM) and *Next Sentence Prediction* (NSP) tasks as training objectives with a slight modification of the NSP task, see Section 3. We evaluate our models on six tasks[4]: Semantic Text Similarity (STS), Named Entity Recognition (NER), Morphological Tagging (MoT), Semantic Role Labeling (SRL), Sentiment Classification (SC) and Multi-label Document Classification (MLC).

Our main contributions are the following ones: 1) We release a pre-trained and ready to use BERT model (Czert-B) for the Czech language that outperforms the compared models on all evaluated sentence-level tasks and it performs comparably on Semantic Role Labeling task. Along with the pre-trained model, we also release the fine-tuned models for each task. 2) We achieve new state-of-the-art results on seven datasets. Moreover we outperform the multilingual models with our newly trained Czert-B model on 7 out of 10 datasets.

2 Related Work

2.1 English BERT and ALBERT

The BERT (Devlin et al., 2018) model adopts the multi-layer Transformer-encoder architecture (Vaswani et al., 2017) with two pre-training tasks: *Masked Language Modeling* and *Next Sentence Prediction*.

The goal of the *MLM* task is to recover artificially distorted sentences where some of the original tokens are *masked out* (hidden), and some are randomly *replaced* with other tokens. These distorted tokens and few other unchanged tokens are selected for prediction (classification). The ratios of predicted tokens can be tuned. For example, in the original BERT model, 15% of input tokens are predicted, 80% of them are masked out, 10% are changed randomly, and 10% are left intact.

The *NSP* is a binary classification task of sentence pairs. For two sentences `A` and `B` taken from the training corpus, the goal is to decide whether the sentence `B` is the actual next sentence (following the sentence `A`) or whether it is a randomly selected sentence from the corpus. In the BERT paper (Devlin et al., 2018), the random sentences are sampled uniformly from the whole corpus.

[4]Some of the evaluation tasks contain more than one independent dataset.

The BERT model represents a big step in massively pre-trained models. The experiments[5] show that a large stack of cross-attention layers with a huge amount of parameters of BERT and BERT-like models can significantly boost the performance of many downstream tasks. A relatively short fine-tuning phase is usually sufficient to set new state-of-the-art results in many tasks using the pre-trained model.

In the original paper (Devlin et al., 2018), the authors publish the $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$ models. $\text{BERT}_{\text{BASE}}$ contains 12 layers, 12 attention heads, and the size of the hidden state is set to 768. In total, it requires 110M parameters. The $\text{BERT}_{\text{LARGE}}$ model has 24 layers, 16 attention heads and the size of the hidden state is set to 1024, which results in 340M parameters.

Training such huge models requires vast computational resources. Therefore, researchers developed methods to reduce the training complexity, memory demands or prediction time, while maintaining similar performance on the fine-tuned tasks. ALBERT model (Lan et al., 2020) represents an example of such an approach.

ALBERT slightly modifies BERT to use the parameters more effectively. First, the authors argue that word embedding size equal to the hidden size (768 for base) is unnecessarily large. They propose to use a smaller size (128) and project the embeddings to the hidden size, which significantly reduces the number of parameters (25M less than in the base variant). Another modification is in cross-layer parameter sharing. In ALBERT, all the weights are shared across all the layers. Another modification consists of replacing the NSP task with a harder task of sentence ordering prediction (SOP). That should result in making the model understand semantics better. The authors introduce models $\text{ALBERT}_{\text{BASE}}$, $\text{ALBERT}_{\text{LARGE}}$, $\text{ALBERT}_{\text{XLARGE}}$, $\text{ALBERT}_{\text{XXLARGE}}$ with 12M, 18M, 60M and 235M parameters, see Table 1.

2.2 BERT-like Models for Other Languages

Researchers publish a multilingual variant of standard $\text{BERT}_{\text{BASE}}$ model (*mBERT*)[6]. It is jointly trained on Wikipedia pages of 104 languages. The model settings are almost the same as in

[5]Experiments in the BERT paper (Devlin et al., 2018) or in many consequent research papers.

[6]See https://github.com/google-research/bert/blob/master/multilingual.md.

$\text{BERT}_{\text{BASE}}$; it differs only in the vocabulary size[7].

However, researchers around the world trained the monolingual variant of the BERT and showed the domination of the monolingual version over the mBERT in many tasks, for example, French (Martin et al., 2020), Finish (Virtanen et al., 2019) or Romanian (Dumitrescu et al., 2020).

Arkhipov et al. (2019) used a combination of four Slavic languages: Bulgarian, Czech, Polish, and Russian. They trained their model using Wikipedia dumps for all four languages and a huge set of Russian news texts. They use the same model architecture and training process as mBERT, and they initialized the model with mBERT weights.

	$\text{BERT}_{\text{BASE}}$	$\text{ALBERT}_{\text{BASE}}$	mBERT	Slavic BERT
Params	$110M$	$12M$	$170M$	$170M$
Vocab size	$40K$	$40K$	$120K$	$120K$
Emb. params	$\approx 30M$	$\approx 5M$	$\approx 90M$	$\approx 90M$

Table 1: Related models parameters.

3 Pre-training Process

3.1 Dataset Description

Training BERT-like models require to collect large quantities of raw text data, pre-process them and prepare automatically labeled training data.

Training corpora We use two publicly available corpora and our crawled dataset of Czech news:

- Czech national corpus (*CsNat*) *28.2GB*, (Křen et al., 2016),
- Czech Wikipedia (*CsWiki*) *0.9GB*, dump[8] from May 2020,
- Crawled of Czech news (*CsNews*), *7.8GB*.

The *CsNat* corpus composes of randomly-ordered blocks of texts sized maximum size of 100 tokens. Each block contains at least one sentence. This must be considered later for the NSP task, which requires a continuous block of texts. Table 3.1 shows the sizes of each corpus in terms of blocks and sentences counts.

Pre-processing We prepare two versions of the corpus: *cased* and *uncased*. Both versions are tokenized with the *WordPiece* tokenizer (Wu et al., 2016) which is trained on the entire corpus.

[7] $\text{BERT}_{\text{BASE}}$ uses a vocabulary with 30K sub-word tokens while mBERT increases the size to 120K tokens.

[8] Taken from `https://dumps.wikimedia.org`

Pre-training Objective We employ MLM and NSP tasks (see section 2.1) for training our model.

The MLM task is used exactly as in the BERT model. The NSP task needs a few considerations. The NSP task requires the availability of continuous blocks of text to form pairs of sentences where one sentence follows the other. At the end of each block, we lose the last sentence that has no sentence to form a pair with. The effect of this issue becomes more apparent with the decreasing length of the continuous text blocks, such as in the case of the *CsNat* corpus. Here, we observe 5.6 sentences per continuous block on average. That means that we are able to use 4.6 sentences out of 5.6 (i.e. approximately 18% of sentences cannot form a pair). When compared to the two remaining corpora, this number is relatively high. In the *CsWiki* and *CsNews* corpora, only 6% and 4%, respectively, of sentences cannot form a pair.

Moreover, we design more difficult negative samples for the NSP task – we select sentences from the same paragraph (that do not directly follow the first sentence) to build non-trivial negative pairs instead of drawing random sentences from the whole corpora as in BERT.

The final dataset consists of 578 158 196 training pairs of sentences. In Table 3.1, we provide some basic statistics of the dataset used in our setup.

	Textual Blocks	Sentences	Avg/block
CsNat	49 104 507	275 314 224	5,61
CsWiki	450 000	6 964 794	15.48
CsNews	2 625 306	58 979 893	22.47

Table 2: Statistics of coropra used.

3.2 Models

We train two models: a smaller $\text{ALBERT}_{\text{BASE}}$ model (Czert-A, 12M parameters) and a larger $\text{BERT}_{\text{BASE}}$ model (Czert-B 110M parameters).

Czert-A is very similar to the standard $\text{ALBERT}_{\text{BASE}}$ with a few modifications: we use WordPiece tokenizer, the batch size is set to 2048 (due to cluster limits), and we use our version of NSP introduced in Section 3 instead of SOP.

Czert-B is configured exactly as the $\text{BERT}_{\text{BASE}}$ model with increased batch size to 2048.

Optimization Both models are trained using a learning rate of 1e-4 with the linear decay using

Adam optimizer (Kingma and Ba, 2014). First, we iterate over the dataset once (single epoch) with the maximum sequence length set to 128. It leads to 300K batches (steps). Similarly to the BERT approach, we then increase the maximum sequence length to 512. We perform about 50K steps with the increased sequence length. In this second shorter iteration, we decrease the batch to 256 samples to fit the cluster memory limits. More details about the computational cluster and its configuration are located in Appendix A.

4 Evaluation

The following section summarizes the performance of Czert on various tasks and compares our model with similar available models. We also add experiments without the pre-training phase to highlight the impact of additional unsupervised data in the Czech language. We also compare Czert with the following baselines:

Baselines

- *SlavicBERT* – a model trained on four Slavic languages (Russian, Bulgarian, Czech and Polish)(Arkhipov et al., 2019),
- *mBERT* – a multilingual version of BERT (Devlin et al., 2018),
- *ALBERT-r* – a randomly initialized ALBERT model without any pre-training.

4.1 Evaluation Tasks

We evaluate our models on six tasks that cover three main groups of NLP tasks: *Sequence Classification* (Sentiment Classification, Multi-label Document Classification); *Sequence Pair Classification* (Semantic Text Similarity); *Token Classification* (Morphological Tagging, Named Entity Recognition, Semantic Role Labeling)

For the *sequence classification* tasks, we take the *pooled* output of the BERT model (and ALBERT). We add dropout and an output layer. The number of output neurons and the activation function differs for each task.

Sentence pair classifications tasks employ the same approach as sequence classification tasks. The only difference is that we feed both sentences separated with special `[SEP]` token together into the model. This way, the model can profit from *cross-attention* between tokens from different sentences.

For the *token classification* tasks, we use the output embeddings associated with the input words (`[CLS]`, `[SEP]` and other special output embeddings are ignored). When the input words are split to sub-word tokens, we take only the first sub-word tokens. For optimization, we use the *Cross-entropy* loss.

For all the tasks, the newly added layers are initialized randomly. We employ the *Adam* optimizer.

4.2 Named Entity Recognition

We use two different datasets to evaluate our model on the named entity recognition task. These are the following:

1. **Czech Named Entity Corpus** (CNEC) (Ševčíková et al., 2007) containing 4 688 training, 577 development and 585 test sentences. We use the CoNLL version of the dataset (Konkol and Konopík, 2013).
2. **BSNLP 2019** shared task dataset (Piskorski et al., 2019) that consists of 196 train and 302 test sentences. We further split the test dataset into development and test parts resulting in development and test datasets of sizes 149 and 153 sentences, respectively. Additionally, we convert the original dataset into the same format as the *CNEC*, extracting entity classes only.

Independently on the dataset, we pre-process the sentences so that the maximum length of an example is 128 sub-word tokens. If the maximum length is exceeded, the residual part is used to create another data point. On the contrary, if the maximum length is not reached, the sentence is padded (padding is inserted at the end of the sentence). It is worth mentioning that exceeding the maximum length of a sentence occurs only for 44 times on the *CNEC*, which is negligible. On the other hand, on the *BSNLP 2019*, the length of the sentences differs a lot, and the maximum length is exceeded for a significant portion of the data. However, our experiments show that increasing the maximum sequence length does not improve the resulting F1 score. The architecture of the model follows the token classification settings described in Section 4.1. See Appendix B.1 for more details about the model and hyper-parameters.

4.2.1 Results

As an evaluation metric, we use F1 score computed on the entity level, while ignoring "O" (empty)

class. The results, stated with 95% confidence intervals, are summarized in Table 3.

	CNEC	BSNLP 2019
mBERT	86.23 ± 0.21	84.01 ± 1.25
SlavicBERT	$\mathbf{86.57 \pm 0.12}$	$\mathbf{86.70 \pm 0.37}$
ALBERT-r	34.64 ± 0.34	19.77 ± 0.94
Czert-A	72.95 ± 0.23	48.86 ± 0.61
Czert-B	86.27 ± 0.12	$\mathbf{86.73 \pm 0.34}$
SoTA	81.77 [b]	**93.9** [a]

Table 3: Comparison of F1 score achieved using pre-trained Czert-A, Czert-B, mBERT, SlavicBERT and randomly initialised ALBERT on NER task. [b]Taken from Konopík and Pražák (2018) [a]Taken from (Arkhipov et al., 2019).

4.3 Morphological Tagging

To evaluate our model on a morphological tagging task, we utilize four Universal Dependencies treebanks. These are namely: Prague Dependency Treebank 3.0 (PDT) (Bejček et al., 2013), Czech Academic Corpus 2.0 (Vildová et al., 2008), Czech Legal Text Treebank 2.0 (Kříž et al., 2018) and FicTree (Hnátková et al., 2017). Together they comprise 103 143 train, 11 326 development and 12 216 test examples. Furthermore, we also perform our experiments on the PDT only to compare our model to the current SoTA. The PDT dataset then comprises 68 627 train, 9 285 dev and 10 163 test examples. The original datasets come as CoNLL files which we converted to a simplified format as in the case of the *CNEC* dataset (section 4.2). During this pre-processing step, we extracted only *UPOS* tags, which we use as labels. The architecture of the model follows the token classification settings described in Section 4.1. The number of output neurons is set to the number of possible *UPOS* tags. See B.2, for more details about the hyper-parameters and training process.

4.3.1 Results

Table 4 shows the achieved results with 95% confidence intervals. Results are stated in F1 score computed on a token level, ignoring the "O" (empty) class. As the table shows, our model *Czert-B* outperforms the other models on both datasets. Moreover, we outperformed the current SoTA (Straka et al., 2019) as well.

	Universal Dependencies	PDT
mBERT	99.176 ± 0.006	99.301 ± 0.005
SlavicBERT	99.211 ± 0.008	99.318 ± 0.008
ALBERT-r	96.590 ± 0.096	96.410 ± 0.060
Czert-A	98.713 ± 0.008	97.028 ± 0.023
Czert-B	$\mathbf{99.300 \pm 0.009}$	$\mathbf{99.410 \pm 0.006}$
SoTA		99.34[a]

Table 4: Comparison of F1 score achieved using pre-trained Czert-A, Czert-B, mBERT, SlavicBERT and randomly initialised ALBERT on morphological tagging task. [a]Result is taken from (Straka et al., 2019).

4.4 Semantic Role Labelling

In semantic role labeling we are looking for shallow semantic structure so the task can be formalized as classification of roles arguments of the predicates in the sentence. Therefore, a single example to be classified is the pair of predicate and argument where the predicate is a single word, and the argument is either word or a phrase. We are classifying the role of the argument towards the predicate. Our input representation is inspired by (Shi and Lin, 2019). We first tokenize the sentence with WordPiece. Then we feed the sentence into the network followed by the `[CLS]` token and the predicate token(s). Note that the predicate tokens have the same positional IDs as their occurrence in the sentence, but different segment ids. This way the predicate at the end of the sequence differs from its in-sentence representation only in segment embedding, so it contains all the information to encode the in-sentence context but it can be easily distinguished from other tokens by the segment embedding.

4.4.1 Results

We evaluate Semantic role labeling for the Czech language on the CoNLL 2009 dataset. The results are shown in Table 5; the *dep-based* column denotes the result achieved by Zhao et al. (2009). In *gold-dep*, we replicated their system but evaluated it with gold-standard dependency trees. Syntax-based F1 metric[9] is computed on whole subtrees of dependency trees. To compute this for span based model, we need to project labels on dependency trees. We did not optimize this projection in any way[10]. We just removed *B-* and *I-* prefixes, we copied the dependency annotation and ran the

[9]Oficial evaluation metric of CoNLL 2009 task.

[10]Because we do not want to add information from gold dependency tree annotations.

	SPAN	SYNTAX
mBERT	78.55 ± 0.11	90.23 ± 0.22
SlavicBERT	79.33 ± 0.08	90.49 ± 0.04
ALBERT-r	51.37 ± 0.42	80.75 ± 0.13
Czert-A	76.63 ± 0.13	89.94 ± 0.05
Czert-B	$\mathbf{81.86 \pm 0.10}$	$\mathbf{91.46 \pm 0.06}$
dep-based	-	85.19
gold-dep	-	89.52

Table 5: SRL results – dep columns are evaluate with labelled F1 from CoNLL 2009 evaluation script, other columns are evaluated with span F1 score same as it was used for NER evaluation.

CoNLL 20009 evaluation script.

As we can see from the table, *Czert-B* and *SlavicBERT* significantly outperform the other models and they even outperform tree-based approach with gold-standard trees. *Czert-B* and *SlavicBERT* performance are very similar in this task.

4.5 Sentiment Classification

Sentiment Classification (SC) task (Liu, 2012) also called *Polarity Detection*, is a classification task where the goal is to assign a sentiment polarity of a given text. The *positive, negative* and *neutral* classes are usually used as the sentiment polarity labels. We perform the evaluation on two Czech sentiment classification datasets from Habernal et al. (2013), consisting of (1) Facebook posts and (2) movie reviews.

The Facebook dataset (*FB*) contains 10K users' posts taken from nine Czech Facebook pages annotated with three[11] classes.

We split the datasets into train, development and test parts with class distribution that follows the original datasets.

We fine-tune the models separately for each dataset. The architecture of the model follows the sequence pair classification setting described in Section 4.1. The number of output neurons is set to the number of sentiment polarity classes. *Softmax* normalization is applied to the output layer. We employ *Cross-entropy* loss. See B.4, for more details about hyper-parameters.

4.5.1 Results

We fine-tune the models (including the baselines) to achieve the best F1 score on the development data. Then, we use the best model settings to train a model on the train and development data. Then, this model is evaluated on the test data and results are reported in Table 6 along with the initial learning rate and the number of epochs used for training. We repeat each experiment six times, and we report the average F1 score along with the 95% confidence interval.

	FB	CSFD
mBERT	71.72 ± 0.91 (2e-5 / 6)	82.80 ± 0.14 (2e-6 / 13)
SlavicBERT	73.87 ± 0.50 (2e-5 / 3)	82.51 ± 0.14 (2e-6 / 12)
ALBERT-r	59.50 ± 0.47 (2e-6 / 14)	75.40 ± 0.18 (2e-6 / 13)
Czert-A	72.47 ± 0.72 (2e-5 / 8)	79.58 ± 0.46 (2e-6 / 8)
Czert-B	$\mathbf{76.55 \pm 0.14}$ (2e-6 / 12)	$\mathbf{84.79 \pm 0.26}$ (2e-5 / 12)
SoTA	69.4[a]	80.5 ± 0.16[b]

Table 6: Average F1 results for the Sentiment Classification task. The numbers in the brackets denote the initial learning rate and number of epochs, respectively, for training of the corresponding model. The state-of-the-art results [a] are taken from (Habernal et al., 2013) and [b] (Sido and Konopík, 2019).

We can see that our Czert-B model outperforms all other models by a large margin on both datasets. We also observe (not shown in the results) for all models that lower initial learning rates (i.e., 2e-6 and 2e-5) lead to more stable fine-tuning than using the initial learning rate of 2.5e-5 which tends to overfit more often as we found out when repeating the experiments. Results for the FB dataset have relatively wide confidence intervals (except for the Czert-B), we believe that it is caused by the small size of the dataset.

4.6 Multi-label Document Classification

Multi-label Document Classification is a variant of classification problem where multiple labels can be assigned to each document. In this problem, there is no constraint on how many of the labels can be assigned to a given document.

We work with the *Czech Text Document Corpus v 1.0* (Hrala and Král, 2013) to fine-tune and evaluate the models. The Czech News Agency provided almost 12 thousands of documents that formed the basis of this dataset. The agency journalists assign 60 categories (tags) to the documents as a part of their daily work. Following the approach from (Lenc and Král, 2018), we use only 37 most frequent categories for evaluation. More statistics are available in the paper.

[11] The dataset contains also 248 samples with a fourth class *bipolar* which we do not use.

	CTDC-1	
	AUROC	F1
mBERT	97.62 ± 0.08	83.04 ± 0.16
SlavicBERT	97.80 ± 0.06	84.08 ± 0.14
ALBERT-r	94.35 ± 0.13	72.44 ± 0.22
Czert-A	97.49 ± 0.07	82.27 ± 0.17
Czert-B	$\mathbf{98.00 \pm 0.04}$	$\mathbf{85.06 \pm 0.11}$
SoTA	–	**84.7**[*]

Table 7: Results for Multi-label Document Classification on Czech Text Document Corpus v 1.0 dataset – AUROC and F1 measures. SoTA taken from (Lenc and Král, 2018).

4.6.1 Model Description and Fine-tuning

For *multi-label classification of documents* (MLC), we follow the sequence classification setting described in Section 4.1. The output layer is activated by the *sigmoid* function. The loss is the *Binary Cross-entropy* function. In the context of this task, documents are regarded as sentences trimmed to the maximum sequence length in tokens set to 512. We chose to pick the first N tokens in each document as our trimming strategy.

We run twenty 10-epoch-long training phases for each model and average the results. See B.6 for more details.

We use both standard *F1* and the *AUROC* (Melo, 2013) evaluation metrics. AUROC represents the overall ability of MLC models to distinguish between different classes without being biased by any constant threshold value. We use 95% confidence interval. We present the results in Table 7.

4.7 Semantic Text Similarity

We evaluate our model on semantic text similarity task on two different datasets.

1. *STS-SVOB* (Svoboda and Brychcín, 2018) contains two datasets: images descriptions (550 train and 300 test samples); and headlines (375 train and 200 test samples). We use the raw variant without any lemmatization or stemming.

2. *STS-CNA* was created during our experiments with this new model in cooperation with Czech News Agency and Charles University. STS-CNA contains s 138,556 hand-annotated sentence pairs (Sido et al., 2021).

	STS-CNA	SVOB-IMG	SVOB-HL
mBERT	90.93 ± 0.34	79.37 ± 0.49	78.83 ± 0.30
SlavicBERT	91.38 ± 0.29	79.90 ± 0.81	77.00 ± 0.31
ALBERT-r	43.18 ± 0.13	15.74 ± 2.99	33.95 ± 1.81
Czert-A	88.72 ± 0.25	79.444 ± 0.34	75.09 ± 0.81
Czert-B	$\mathbf{91.89 \pm 0.12}$	$\mathbf{83.74 \pm 0.40}$	$\mathbf{79.83 \pm 0.47}$
SoTA[*]	–	*78.87*	**79.99**

Table 8: Pearson correlation (95% conf. from ten experiments). [*]Taken from Svoboda and Brychcín (2018)

	CNA	SVOB-IMG	STS-SVOB-HL
mBERT	87.88 ± 0.08	78.83 ± 0.36	$\mathbf{78.83 \pm 0.37}$
SlavicBERT	88.97 ± 0.09	79.66 ± 0.73	76.03 ± 0.42
ALBERT-r	33.32 ± 0.11	15.15 ± 3.07	32.25 ± 2.05
Czert-A	85.85 ± 0.16	78.72 ± 0.38	73.86 ± 0.72
Czert-B	$\mathbf{89.29 \pm 0.17}$	$\mathbf{83.20 \pm 0.39}$	$\mathbf{78.69 \pm 0.59}$

Table 9: Spearman correlation (95% conf. from ten experiments)

4.7.1 Model Description and Fine-tuning

The architecture of the model follows the sequence pair classification setting described in Section 4.1. The number of output neurons is set to 1, and no activation function is applied to the output layer. We employ the *Mean Squared Error* loss.

We tried to keep hyper-parameters as close as possible between all experiments; however, we were forced to change them slightly in case of Czert-A and ALBERT-r. Also, the datasets have different nature; thus, we use different sets of hyper-parameters for each dataset. See B.5

We run ten experiments for each configuration and use 95% confidence interval. The tables Table 8 and Table 9 summarize the results. Table 8 shows that Czert-B model significantly outperforms the SoTA on SVOB-IMG dataset. In the SVOB-HL dataset, the models perform in par. We believe that the draw can be caused by reaching the annotation accuracy limit of this dataset.

We also observe a more stable and robust training on extremely small datasets; both Czert models are less prone to over-fitting than other tested models.

5 Discussion

We summarize the overall results of all evaluated tasks in Table 10. The first three columns contain the token classification tasks, the next two columns show results for sequence classification tasks, and the last column belongs to sequence pair classification task. We can observe that Czert-B model

	NER		MoT		SRL	SENTIMENT		MULTI-CLASS	STS		
	CENEC	BSNLP	UNIV. DEP.	PDT	CoNLL-09	FB	CSFD	CTDC-1	CNA	SVOB-IMG	SVOB-HL
mBERT	86.23	84.01	99.176	99.301	90.23	71.72	81.35	83.04	90.93	79.37	78.83
SlavicBERT	**86.57**†	86.70	99.211	99.318	90.49	73.87	81.55	84.08	91.38	79.90	77.00
ALBERT-r	34.64	19.77	96.590	96.410	80.75	59.50	70.33	72.44	43.18	15.73	33.95
Czert-A	72.95	48.86	98.713	97.028	89.94	72.47	79.73	82.27	88.72	79.44	75.09
Czert-B	86.27	86.73	**99.300**†	**99.410**†	**91.46**†	**76.55**†	**84.79**†	**85.06**†	**91.89**†	**83.74** †	**79.83**
SoTA*	81.77	**93.9**	–	99.34	89.52	69.4	80.5	84.7	–	78.87	**79.99**

Table 10: Summary of our results. The bold results denote the current SoTA results. The underlined results are the best result achieved directly by fine-tuning the BERT-like models. Values with the † symbols are the new SoTA results that we established in this paper. *Results are taken from original papers.

excels at the sequence and sequence pair classification tasks. In these tasks, Czert-B outperforms other pre-trained models by a large margin. We believe that the likely cause for such results lay in the amount of Czech data we use to train Czert models. mBERT and SlavicBERT use only Czech Wikipedia, but we work with almost 50 times larger data in terms of sentence count. For most of the token classification tasks, Czert-B performs similarly to other pre-trained models except for SRL, where Czert-B outperformed other models by a large margin.

We establish a new state of the art on **NER** with the SlavicBERT model on the CNEC dataset. The performance increase is a major one. We increase the F1 measure by 5%. Also, we achieve similar results with SlavicBERT and Czert-B on BSNLP dataset.

We also outperformed other BERT-like models with Czert-B in **MoT**, and surpass the SoTA.

We accomplish outstanding performance and increase the SoTA in two other tasks: sentiment classification (**SC**) and semantic text similarity (**STS**). The increase is of ∼5% and ∼3% in both sentiment datasets and of ∼5% in one of the semantic similarity datasets. We also overcame SoTA in **MLC**.

6 Conclusion

In this work, we present two monolingual BERT-like models (BERT and ALBERT) for the Czech language. We train the models with the original MLM task and with a slightly modified NSP task. We thoroughly evaluate our models on six common tasks, and we compare them with other multilingual models. We include task-specific state-of-the-art models in our comparison. We outperform multilingual models with our newly trained Czert-B model on 9 out of 11 datasets. In addition, we establish the new state-of-the-art results on 9 datasets[12]. The results show the strong performance of the Czert-B model on STS, MLC, SC, SRL, and MoT tasks. As our paper confirms and as is shown in similar works, monolingual Transformer-based language models often overcome the multilingual ones.

Our models are publicly available for research purposes at our website and in the hugging face repository[13].

Acknowledgement

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

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

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,

[12]The results in Table 10 with the † symbol.

[13]https://huggingface.co/UWB-AIR

Florence, Italy. Association for Computational Linguistics.

Eduard Bejček, Eva Hajičová, Jan Hajič, Pavlína Jínová, Václava Kettnerová, Veronika Kolářová, Marie Mikulová, Jiří Mírovský, Anna Nedoluzhko, Jarmila Panevová, Lucie Poláková, Magda Ševčíková, Jan Štěpánek, and Šárka Zikánová. 2013. Prague dependency treebank 3.0.

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.

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, pages 7059–7069. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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.

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.

Milena Hnátková, Tomá Jelínek, Ivana Klímová, Alena Kropíková, Hana Skoumalová, Olga Zitová, and Daniel Zeman. 2017. Fictree.

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.

M. Hrala and P. Král. 2013. Evaluation of the document classification approaches. In *8th International Conference on Computer Recognition Systems (CORES 2013)*, pages 877–885, Milkow, Poland. Springer.

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

Michal Konkol and Miloslav Konopík. 2013. Crf-based czech named entity recognizer and consolidation of czech ner research. In *Text, Speech and Dialogue*, volume 8082 of *Lecture Notes in Computer Science*, pages 153–160. Springer Berlin Heidelberg.

Miloslav Konopík and Ondřej Pražák. 2018. Lda in character-lstm-crf named entity recognition. In *International Conference on Text, Speech, and Dialogue*, pages 58–66, Cham. Springer International Publishing.

Michal Křen, Václav Cvrček, Tomáš Čapka, Anna Čermáková, Milena Hnátková, Lucie Chlumská, Tomáš Jelínek, Dominika Kováříková, Vladimír Petkevič, Pavel Procházka, Hana Skoumalová, Michal Škrabal, Petr Truneček, Pavel Vondřička, and Adrian Zasina. 2016. SYN v4: large corpus of written czech. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Vincent Kříž, Barbora Hladká, and Zdeňka Urešová. 2018. Czech legal text treebank 2.0.

Dariusz Kłeczek. 2020. Polbert: Attacking polish nlp tasks with transformers. In *Proceedings of the PolEval 2020 Workshop*. Institute of Computer Science, Polish Academy of Sciences.

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*.

Ladislav Lenc and Pavel Král. 2018. Deep neural networks for Czech multi-label document classification. In *Computational Linguistics and Intelligent Text Processing*, pages 460–471, Cham. Springer International Publishing.

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

Y Liu, M Ott, N Goyal, J Du, M Joshi, D Chen, O Levy, M Lewis, L Zettlemoyer, and V Stoyanov. Roberta: A robustly optimized bert pretraining approach. arxiv 2019. *arXiv preprint arXiv:1907.11692*.

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, pages 6294–6305. Curran Associates, Inc.

Francisco Melo. 2013. *Area under the ROC Curve*, pages 38–39. Springer New York, New York, NY.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26, pages 3111–3119. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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.

Jakub Piskorski, Laska Laskova, Michał Marcińczuk, Lidia Pivovarova, Pavel Přibáň, Josef Steinberger, and Roman Yangarber. 2019. The second cross-lingual challenge on recognition, normalization, classification, and linking of named entities across Slavic languages. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 63–74, Florence, Italy. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Stefan Schweter. 2020. Berturk - bert models for turkish.

Alexander Sergeev and Mike Del Balso. 2018. Horovod: fast and easy distributed deep learning in TensorFlow. *arXiv preprint arXiv:1802.05799*.

Magda Ševčíková, Zdeněk Žabokrtský, and Oldřich Krůza. 2007. Named entities in czech: Annotating data and developing NE tagger. In *Lecture Notes in Artificial Intelligence, Proceedings of the 10th International Conference on Text, Speech and Dialogue*, volume 4629 of *Lecture Notes in Computer Science*, pages 188–195, Berlin / Heidelberg. Springer.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

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

Jakub Sido, Michal Seják, Ondřej Pražák, Miloslav Konopík, and Václav Moravec. 2021. Czech news dataset for semanic textual similarity. *arXiv preprint arXiv:2108.08708*.

Milan Straka, Jana Straková, and Jan Hajič. 2019. Czech text processing with contextual embeddings: Pos tagging, lemmatization, parsing and ner. In *International Conference on Text, Speech, and Dialogue*, pages 137–150. Springer.

Lukás Svoboda and Tomás Brychcín. 2018. Czech dataset for semantic textual similarity. In *Text, Speech, and Dialogue - 21st International Conference, TSD 2018, Brno, Czech Republic, September 11-14, 2018, Proceedings*, volume 11107 of *Lecture Notes in Computer Science*, pages 213–221. Springer.

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.

Barbora Hladká Vildová, Jan Hajič, Jiří Hana, Jaroslava Hlaváčová, Jiří Mírovský, and Jan Raab. 2008. Czech academic corpus 2.0.

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]*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763. Curran Associates, Inc.

Hai Zhao, Wenliang Chen, Chunyu Kit, and Guodong Zhou. 2009. Multilingual dependency learning: A huge feature engineering method to semantic dependency parsing. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 55–60.

A Cluster Configuration

We use distributed training to set the weights of Czert. For distributed pre-training we rely on the Czech national cluster Metacentrum[14]. We employ 16 machines, each with two NVIDIA TESLA T4 graphic cards, which results in 32 T4s in total.

For the Czert-A model, we use standard Tensorflow (Abadi et al., 2015) distributed training, which is based upon the gRPC standard. It takes 12 days to training Czert-A with this setting.

The Czert-B model contains almost ten times as many trainable parameters as the Czert-A model. It proved impractical to train Czert-A with the tools provided by Tensorflow alone. We employ the MPI messaging standard that communicates over the OmniPath network with a speed of 100Gb/s. The Horovod (Sergeev and Balso, 2018) library handles all the synchronization transfers of our distributed training. We are able to reach the speeds of 2400ms per batch with this setting, which is approximately five times faster than with standard gRPC via TCP/IP. We are able to train the Czert-B model in 8 days.

B Fine-tuning and Hyper-parameters

B.1 Named Entity Recognition

In all of our experiments, we use Adam optimizer with a learning rate of 5e-5 and a linear decay to zero. Additionally, the Czert-B model uses a learning rate warm-up during the first epoch. All the models are trained with batch size 64 for 25 epochs on an NVIDIA Tesla-T4 GPU. For Czert-A it takes approximately 25 minutes on the *CNEC* dataset, whereas on the *BSNLP 2019* it takes less than 7 minutes.

B.2 Morphological Tagging

The architecture of the model follows the token classification setting described in Section 4.1. The number of output neurons is set to the number of morphological tags in Universal Dependencies. Namely:

- Prague Dependency Treebank 3.0,
- Czech Academic Corpus 2.0,
- Czech Legal Text Treebank 2.0,
- FicTree.

For fine-tuning, we use Adam optimizer with a learning rate of 5e-5 and a linear decay to zero. Additionally, the Czert-B model uses a learning rate warm-up during the first epoch. Similarly to our NER experiments (Section 4.2s), we use a maximum sequence length of 128 sub-word tokens. The models are trained with batch size 64 for 13 epochs. For Czert-A it takes about 8 hours and 15 minutes on an NVIDIA Tesla-T4 GPU.

B.3 Semantic Role Labeling

For fine-tuning, we use Adam optimizer with a learning rate of 5e-5 and a linear decay to zero. We use a maximum sequence length of 128 sub-word tokens. We train the model on 2 Tesla T4 graphic cards with batch size of 64 for 12 epochs.

B.4 Sentiment Classification

We perform fine-tune training of the models by minimizing the Cross-Entropy loss function using the Adam (Kingma and Ba, 2014) optimization algorithm with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) and with a linear learning rate decay (without warm-up). We try three different initial learning rates, i.e., 2e-6, 2e-5 and 2.5e-5 for at most 14 epochs. We use a max sequence length of 64, batch size of 32 for the FB[15] dataset and a max sequence length of 512 and batch size of 14 for the CSFD dataset.

B.5 Semantic Textual Similarity

For the CNA dataset, we train two epochs using a batch of size 50, and LR 1e-5 with linear decay to zero for each model except Czert-A for which we used 5e-6 for four epochs, which lead to slightly better results.

For smaller datasets (SVOB-img and SVOB-hl) we used LR 5e-6 and train on 14k batches.

For each experiment, we used Adam optimizer, L2 weight normalization, and learning rate warm-up during the first 500 batches.

B.6 Multi-label Document Classification

For each experiment, we first run a linear grid search through learning rate parameter L = {2e-5, 4e-5, ..., 10e-4} and a decision $D = \{true, false\}$ whether to use a linear learning rate decay[16] or to

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

[15] Even though that we use different tokenizers for each model, number of tokens in posts from the FB dataset do not exceed 66 tokens and average number of tokens around 20 for all tokenizers.

[16] Arriving at 0 at the end of the last epoch.

keep the maximum learning rate constant until the last step. The learning rate achieved maximum after 500 steps of the warm-up phase. After the grid search was complete, we've run twenty 10-epoch-long training phases for each of the extended models and average the results.