RoBERT – A Romanian BERT Model

Mihai Masala **Stefan Ruseti** **Mihai Dascalu**
University Politehnica of Bucharest
313 Splaiul Independentei
060042 Bucharest, Romania
mihaimasala@gmail.com {stefan.ruseti, mihai.dascalu}@upb.ro

Abstract

Deep pre-trained language models tend to become ubiquitous in the field of Natural Language Processing (NLP). These models learn contextualized representations by using a huge amount of unlabeled text data and obtain state of the art results on a multitude of NLP tasks, by enabling efficient transfer learning. For other languages besides English, there are limited options of such models, most of which are trained only on multi-lingual corpora. In this paper we introduce a Romanian-only pre-trained BERT model – RoBERT – and compare it with different multi-lingual models on seven Romanian specific NLP tasks grouped into three categories, namely: sentiment analysis, dialect and cross-dialect topic identification, and diacritics restoration. Our model surpasses the multi-lingual models, as well as a another mono-lingual implementation of BERT, on all tasks.

1 Introduction

In the last decade, Recurrent Neural Networks (RNNs) based on LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Chung et al., 2014) cells represented the basis of state of the art methods for a wide range of Natural Language Processing (NLP) tasks (Bahdanau et al., 2015; Wang and Tan, 2016; Mehri and Carenini, 2017; Wang and Jiang, 2017). In general, RNNs make great use of pre-trained word embeddings such as word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014). Word embeddings are usually computed using specialized neural networks trained in an unsupervised manner, and learn for each word a single vector representation. Recently, a paradigm-shift in the NLP community occurred: word embeddings were replaced by large-scale pre-trained language models that compute contextual embeddings (i.e., output embeddings depend on the entire sequence). Transformer (Vaswani et al., 2017) has quickly become the building block of multiple state of the art architectures such as GPT (Radford et al., 2018; Radford et al., 2019; Brown et al., 2020), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), or XLNET (Yang et al., 2019). Vaswani et al. (2017) propose the usage of multiple-head self-attention blocks, instead of the more classical recurrent approach, to model long-range sequence interactions. Replacing the sequential recurrent neural network with self-attention modules allows for easy parallelization, thus ensuring faster training on large-scale architectures. Large-scale transformers have the advantage of a single computationally expensive phase (pre-training), followed by an easy and fast fine-tuning phase, specific for each task.

While transformer models have quickly become the standard approach for NLP tasks, the vast majority of studies have been performed on English. For other languages, the options are rather limited: either pre-train an entire model on the preferred language, or use a multi-lingual model trained on several languages. Two multi-lingual models stand out: multi-lingual BERT (Devlin et al., 2019), which is a BERT-base model trained on 104 languages, and XLM-RoBERTa (Conneau et al., 2020), which is trained on a massive 2.5TB corpus containing samples from 100 languages.

In this paper, we set out to pre-train BERT-based models for Romanian and perform an extensive study on its performance on a multitude of downstream tasks. Three variants of RoBERT (i.e. small, base, and

Proceedings of the 28th International Conference on Computational Linguistics, pages 6626–6637
Barcelona, Spain (Online), December 8-13, 2020

large) were pre-trained and publicly released at [1]. Both multi-lingual and mono-lingual (Romanian) BERT models were analyzed and compared on three downstream tasks (one with five sub-tasks), showing that RoBERT variants consistently outperform multi-lingual models and previous approaches on all considered tasks.

2 Related Work

A study by Rönnqvist et al. (2019) compared mono-lingual variants of BERT on English and German, with multi-lingual BERT. Experiments were conducted on a simple syntactic classification task, a cloze test, and full text generation. On the simple syntactic classification task, rather small differences were encountered between mono- and multi-lingual models. As the tasks increased in difficulty, the gap between model performance increased to the point where multi-lingual BERT was barely usable for language generation. The study concludes that a real need exists for mono-lingual BERT models, instead of relying on multi-lingual ones.

Mono-lingual variants of BERT-based models are available for a multitude of languages. FlauBERT (Le et al., 2020) and CamemBERT (Martin et al., 2020) are both RoBERTa-based (Liu et al., 2019) French models, that outperform the multi-lingual variant of BERT on a variety of NLP tasks. Dutch versions of BERT (de Vries et al., 2019; Delobelle et al., 2020) are also available, as well as models for Russian (Kuratov and Arkhipov, 2019), German[2], Finnish (Virtanen et al., 2019), Italian (Polignano et al., 2019), Portuguese (Souza et al., 2019), Spanish (Cañete et al., 2020), Vietnamese (Nguyen and Nguyen, 2020), Japanese[3], and Arabic (Antoun et al., 2020).

There are only two multi-lingual BERT-based models available for the Romanian language at the time of writing this paper, namely mBERT (Devlin et al., 2019) and the more recent XLM-RoBERTa (XLM-R) (Conneau et al., 2020). We only found one repository[4] with a model trained specifically for Romanian. Unfortunately, we did not find a very great level of details regarding how their model was trained. A great overlap between our and their pre-training corpora exists, although the collections are not identical (i.e., both approaches are mostly based on Oscar (Javier Ortiz Suárez et al., 2019) and Romanian Wikipedia; more details in Section 3.1). One noteworthy difference is represented by the size of the vocabulary, 38k tokens (ours) and 50k tokens (theirs). In all following experiments, we refer to this model as *BERT-base-ro*. To our knowledge, these models are the only Transformer-based options for Romanian.

3 Building RoBERT

3.1 Corpus

A large Romanian corpus extracted from multiple sources was built for pretraining RoBERT, ranging from random text crawled from the Internet, to more formal sources (e.g. Wikipedia, books or newspapers). The corpus was compiled from three main sources: a Romanian Wikipedia dump[5], a Romanian corpus provided by Oscar (Javier Ortiz Suárez et al., 2019), together with online Romanian sources selected from the RoTex collection[6]). Details on the number of words, sentences, and uncompressed sizes are available in Table 1. Out of the entire corpus, 10% was set aside to be used as an evaluation corpus for our models. Details of dataset sizes used for different Transformers-based architectures are presented in Table 2. Despite using the same dataset, Yang et al. (2019) report the size of BERT as 13 GB, while Liu et al. (2019) report 16GB. This difference is probably due to slightly different cleaning mechanisms. Lan et al. (2020) experiment for ALBERT with both the original dataset used for training BERT (Devlin

[1] https://git.readerbench.com/ReaderBench/readerbenchpy#how-to-use-bert

[2] https://deepset.ai/german-bert

[3] https://github.com/cl-tohoku/bert-japanese

[4] https://github.com/dumitrescustefan/Romanian-Transformers

[5] from 2019-08-01

[6] https://github.com/aleris/ReadME-RoTex-Corpus-Builder from which the following sources were considered: ”biblior”, ”biblioteca-digitala-ase”, ”bestseller-md”, ”litera-net”, ”bzi”, ”dcep”, ”dezbateri-parlamentare”, ”dgt-aquis”, ”paul-goma”, ”rudolf-steiner” and ”ziarul-lumina”

et al., 2019), as well as datasets used for training RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019).

Data source	Words	Sentences	Size (GB)
RoWiki	50M	2M	0.3
Oscar (Javier Ortiz Suárez et al., 2019)	1.78B	87M	10.8
RoTex	240M	14M	1.5
Total	2.07B	103M	12.6

Table 1: Statistics of cleaned data sources used for pre-training RoBERT.

Model	Training dataset size (GB)
BERT (Devlin et al., 2019)	13
RoBERTa (Liu et al., 2019)	160
XLNet (Yang et al., 2019)	130
ALBERT (Lan et al., 2020)	13/130/160
RoBERT (ours)	12.6

Table 2: Details on dataset sizes for different Transformers-based architectures.

3.2 Model architecture

RoBERT model architecture is based on a multi-layer bidirectional Transformer (Vaswani et al., 2017), similar to BERT. Devlin et al (2019) propose two configurations for the Transformer, namely BERT-base and BERT-large. We propose the following configurations for RoBERT: RoBERT-small, RoBERT-base, and RoBERT-large (see Table 3). It is important to note that RoBERT-base and RoBERT-large follow the exact layer sizes as BERT-base and BERT-large, respectively.

Model	Weights	Vocab size	L	H	A	Training Time (h)*
mBERT	177M	120k	12	768	12	-
XLM-R-base	278M	250k+	12	768	12	-
BERT-base-ro	124M	50k	12	768	12	-
RoBERT-small	19M	38k	12	256	8	28
RoBERT-base	114M	38k	12	768	12	77
RoBERT-large	341M	38k	24	1024	16	255

* On a TPU v3-8

+ XLM-R uses Sentence Piece tokenization instead of Word Piece

Table 3: Number of trainable weights in millions, vocabulary tokens, number of Layers (L), Hidden size (H), and number of Attention heads (A) for each considered model.

3.3 Model training

We closely follow the same methodology proposed by Devlin et al. (2019) for training our models. The original BERT model was trained using two supervised tasks: *masked language model* (MLM) in which the model is trained to predict randomly masked tokens, and *next sentence prediction* (NSP) in which the model learns whether two sentences follow each other or are randomly sampled from the training dataset. The usefulness of the NSP task is still debatable: Devlin et al. (2019) showed that using the NSP task increases performance, while others have shown that the NSP task actually hinders performance for slightly modified Transformers architectures (Liu et al., 2019; Yang et al., 2019). Lan et al. (2020) introduce a sentence order prediction (SOP) task in which the model has to predict whether two sentences are given in the correct order or are reversed. We decided to use both MLM and NSP objectives.

We followed the approach proposed by Devlin et al. (2019) to optimize the objective function with the following hyperparameters: Adam optimizer (Kingma and Ba, 2015) with a learning rate of 1e-4, β_1 of 0.9, β_2 of 0.999, L_2 decay of 0.01, linear decay of the learning rate, and learning rate warmup over the first 1% of the training steps.

The models were trained for 40 epochs, on a v3-8 TPU (provided by TensorFlow Research Cloud[7]) with the maximum batch size that fits into the memory. Because the attention mechanism has quadratic complexity in relation to the sequence length, 90% of the steps were trained with a maximum sequence size of 128, while for the rest of 10% a maximum sequence length of 512 was used. Training with sequences of length 512 is needed to learn all positional embeddings. Devlin et al. (2019) used the same approach for training BERT.

In addition, small adaptations were made to the tokenization process to take into account diacritics, as they are important for the Romanian language. All our models share the same WordPiece vocabulary of 37,788 tokens.

4 Evaluation

In the following section, we describe the methodology, tasks, and datasets used to evaluate and compare our models with other state-of-the-art methods applicable for the Romanian language. Unfortunately, we are not aware of any large collection of natural language understanding tasks for Romanian language. Therefore, the models were tested on a total of seven downstream tasks grouped into three categories: sentiment analysis, Moldavian versus Romanian dialect and cross-dialect topic identification (with five sub-tasks), and automated diacritics restoration. We believe the seven tasks represent a reliable benchmark for comparing natural language understanding models because they are well balanced, having data sources originating from various informal (i.e. online product reviews), semi-formal (i.e. talk show scripts), and formal (i.e. news) sources. Furthermore, we present MLM and NSP accuracies and losses computed on the evaluation corpus (see Table 4).

Model	MLM loss	MLM accuracy	NSP loss	NSP accuracy
RoBERT-small	2.4576	0.5363	0.0838	0.9687
RoBERT-base	1.7073	0.6511	0.0601	0.9802
RoBERT-large	1.4578	0.6929	0.0444	0.9843

Table 4: Performance on the evaluation set

4.1 Sentiment analysis

A corpus was required to test the capability of our models to capture and classify sentiments. Thus, around 160k Romanian reviews were crawled from one of the most popular online shopping platforms in Romania, namely *eMAG*[8]. Reviews covered 129 distinct product categories, which can be summarized into six main categories: 1) IT (e.g., notebooks, computer parts) - 44%, 2) electronics (home appliances) - 23%, 3) fashion and personal care products - 15%, 4) tools - 7%, 5) car accessories - 6%, and 6) other (products that cannot be categorized into any previous category) - 5%. The review content written by the customer and its associated score (stars between 1 and 5) are considered. Although more information is available (such as review title, review date, product name, product category, and product description), this analysis relies only on basic information: the review body and its score. This decision was made

[7] https://www.tensorflow.org/tfrc

[8] https://www.emag.ro/

because the goal of the following experiments is to test and compare our models directly with multilingual BERT. Thus, we opted to make the final model architecture as simple as possible. Adding a larger or more diverse set of features could diminish the differences between the models.

The dataset is greatly unbalanced, as people tend to either write a positive or a negative review, and rarely express a more balanced or neutral review. Our crawled dataset contains 2.5 times more reviews of 1 or 5 stars than other reviews (2, 3 or 4 stars). To alleviate this issue, two different strategies were employed: reducing the number of classes from 5 to 4 by combining the reviews of 2 and 3 stars into a single class, and performing under-sampling when training the model. A classic train/dev/test split is used, having 0.8/0.1/0.1 proportions with a stratified approach.

The final dataset for this task contains 4 classes with about 133k reviews for train and 16k for dev and test, respectively. We decided to undersample the majority class, namely to select only 20k samples (out of the 85k reviews) from the class containing 5 star reviews (the majority class), leaving us with a balanced training dataset.

4.2 Moldavian vs. Romanian Dialect and Cross-dialect Topic identification

The Moldavian and Romanian Dialectal Corpus (MOROCO) (Butnaru and Ionescu, 2019) is a large dataset containing over 30k samples of Romanian and Moldavian texts crawled from online news sources. Each sample from the corpus is annotated with both dialectal and category information. This enables the definition of five different tasks: binary classification by dialect (discriminate between Romanian and Moldavian dialects), two intra-dialect classification tasks, and two cross-dialect classification tasks. Each text sample comes from one of six news categories: culture, finance, politics, science, sports, or tech. For the two intra-dialect classification tasks, the model is trained on either Romanian or Moldavian text samples, and evaluated on the same dialect, leading to Romanian (RO) topic classification and Moldavian (MD) topic classification, respectively. Cross-dialect classification tasks imply training the model on one of the dialects and evaluating its performance on the other (i.e, training on Romanian samples and testing on Moldavian samples, known as MD to RO categorization, followed by RO to MD categorization).

The MOROCO dataset was also used at the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2019[9]) in the form of Moldavian vs. Romanian Cross-dialect Topic identification (MRC). The challenge contained three of the five original tasks (i.e., binary classification by dialect and two cross-dialect classification tasks). The training set for each of the three tasks was the same as in the MOROCO dataset, the development set contained both dev and test sets from the original MOROCO, while the test set was new, distinct, and private.

4.3 Diacritics restoration

Diacritics restoration is the task of processing a text without diacritics and adding them, where required. In Romanian, there are four characters that can accept diacritics (i.e. *a*, *i*, *s*, and *t*), leading to the following set of characters: *ă*, *â*, *î*, *ș*, and *ț*.

Iordache et al. (2019) introduce a free and large scale dataset containing over 40M words and over 2.5M sentences tailored for automated diacritics restoration. The sources used for building the dataset include online news and talk-shows scripts. The corpus is preprocessed, cleaned, and split into train, validation and test set. Furthermore, the gold-standard for the test is private, but there is a public challenge[10], together with a leaderboard made available by the authors.

5 Experiments and Results

5.1 Experimental setup

A standard approach is followed for all experiments: train the models on the training dataset for a number of epochs, select the model with the best performance on the development set, and run that model on the test set. Several metrics were reported for all models on both the development and test sets. More training

[9] https://sites.google.com/view/vardial2019

[10] http://diacritics-challenge.speed.pub.ro/

details (e.g. number of epochs, learning rate) are presented in each section independently, followed by results and discussions.

5.2 Sentiment analysis

Two fully connected layers with 100 and 4 units respectively were added for the sentiment analysis task on top of the ”CLS” representation computed by the BERT-based models. The entire architecture was fine-tuned for a maximum of 10 epochs. Each batch contains 64 examples, and cross-entropy loss is reduced by using the Adam optimizer (Kingma and Ba, 2015), with a learning rate of 1e-5. A grid search was performed for establishing the optimal maximum sequence length (256 and 512) and dropout rate (0.1 and 0.5). In early testing, no significant performance differences were noticed when experimenting with other learning rates (5e-5, 2e-5 or 1e-5). The best model was selected based on the macro F1 metric performance on the development set. A smaller batch size was used for the XLM-R model due to computational limits.

The results obtained on the sentiment analysis task are presented in Table 5. The first section of Table 5 introduces the performance of the baseline models, namely multi-lingual BERT, XLM-R-base and BERT-base-ro, followed by all three variants of RoBERT in the second section. Our base and large models outperform mBERT across the board on both the development and test set. XLM-R-base obtains better overall performance than all ”base” models, but obtains only a 0.1 F1 score improvement (71.71 vs 71.61) over RoBERT-base, while having more than 2x the number of parameters (278M versus 114M). We also note a small performance difference between RoBERT-base and BERT-base-ro, in favor of our model, a difference that is consistent across all considered metrics. While considering only RoBERT models, we observe that increasing the model size yields better performance on all metrics, RoBERT-large obtaining the best scores out of all considered models.

Model	**Performance on dev set**			**Performance on test set**		
	accuracy	**macro F1**	**weighted F1**	**accuracy**	**macro F1**	**weighted F1**
mBERT	76.97	68.96	78.21	77.54	69.57	78.71
XLM-R-base	78.41	71.26	79.70	**79.19**	71.71	80.37
BERT-base-ro	77.69	70.49	79.07	78.17	71.02	79.43
RoBERT-small	76.13	66.32	77.23	76.25	66.37	77.29
RoBERT-base	77.77	70.89	79.32	78.32	71.61	79.78
RoBERT-large	**79.22**	**72.48**	**80.51**	79.16	**72.11**	**80.44**

Table 5: Accuracy, macro-averaged F1 and weighted F1 scores (in %) for the sentiment analysis task.

5.3 Moldavian vs. Romanian Dialect and Cross-dialect Topic identification

For all subsequent tasks, a similar approach to the previous one is considered. Two fully connected layer with sizes of 100 and the number of classes for each task (i.e., two for binary classification by dialect, and six for the other tasks) are added on top of the ”CLS” token representation. The following hyperparameters are used: Adam optimizer (Kingma and Ba, 2015), a learning rate of 1e-5, and a batch size of 64.

Table 6 introduces the results for all sub-tasks of the MOROCO dataset, on both the development set and the test sets. The baseline introduced by Butnaru and Ionescu (2019) is also presented - their model uses Kernel Ridge Regression on top of features extracted by String Kernels (this model is future referenced as KRR + SK). Note that for the cross-dialect topic classification tasks, the training and dev sets are from the same dialect, following the approach proposed by Butnaru and Ionescu (2019).

For the binary dialect classification task, even our smallest model (i.e., RoBERT-small) outperforms all considered baselines on the development set, and is just slightly below BERT-base-ro on the test set. Both RoBERT-base and RoBERT-large add at least 1.0% F1 score to any of the considered baselines.

While increasing model size achieves better performance, going beyond the ”base” variant does not seem to yield any noteworthy difference (at least on the test set; we will later observe a different behaviour on the VarDial 2019 challenge).

Our RoBERT-base outperforms all considered baselines on both development and test set for both intra-dialect topic classification tasks. On Moldavian topic classification, the usage of a larger model (i.e. RoBERT-large) does not increase the performance on the test set, a phenomenon also observed on the binary classification task. This is not the case for the Romanian topic classification where an increase in performance with RoBERT-large is observed.

In the case of cross-dialect tasks, namely MD to RO and RO to MD topic classification, the model is trained on one dialect, and evaluated on the test set from the other dialect. No additional architecture adjustments were made to take into account cross-domain training and evaluation. Therefore, the BERT based models are ”as-is”, they are not pre-trained in any manner for cross-domain tasks. A better performance on the development set (in Moldavian dialect) for the MD to RO topic classification task does not directly translate into a better performance on the test set (in Romanian). Actually, out of all BERT-based models, one of the worst performing model on Moldavian (XLM-R-base with macro F1 92.45 on development set) obtains the best performance on Romanian (70.57 F1 macro on test set). Nevertheless, we observe a similar phenomenon on the RO to MD topic classification task to the previous task: RoBERT-base and RoBERT-large perform better than all considered baselines, RoBERT-large obtaining state-of-the-art results.

Model	**Dialect**		**MD Topic**		**RO Topic**		**MD to RO**		**RO to MD**	
	Dev	**Test**	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**
KRR + SK	94.15	94.06	90.45	90.57	77.68	78.76	90.45	67.59	77.68	75.47
mBERT	95.29	95.31	92.56	91.66	83.30	82.11	92.56	68.95	83.30	78.59
XLM-R-base	96.12	95.61	92.45	91.63	83.86	82.50	92.45	**70.57**	83.86	80.81
BERT-base-ro	95.58	95.98	92.37	91.45	83.12	82.77	92.37	69.90	83.12	78.08
RoBERT-small	96.00	95.76	92.73	91.45	83.50	81.86	92.73	69.05	83.50	80.15
RoBERT-base	97.36	**97.24**	93.92	**93.40**	83.95	82.93	93.92	68.80	83.95	82.37
RoBERT-large	**97.68**	97.21	**94.82**	93.26	**84.36**	**83.24**	**94.82**	69.50	**84.36**	**83.26**

Table 6: Macro-averaged F1 score (in %) for a) dialect classification (RO vs MD), b) two intra-dialect topic classifications (MD Topic and RO Topic), and c) two inter-dialect topic classifications (MD to RO and RO to MD) on the MOROCO dataset.

In addition to the experiments performed on the original MOROCO dataset, Table 7 introduces the results on VarDial MRC. The best model in the competition is introduced for all three tasks, together with the best post-competition results (in parenthesis). Tudoreanu (2019) proposed an approach based on skip-gram convolutional neural networks (CNN), and a CNN trained on triplets (anchor, positive and negative sample) combined using Support Vector Machines (SVM) - this model is future referred as 2-CNN + SVM. Wu and Kwok (2019) used SVMs with character and ngram features weighted with Tf-Idf BM25 weighting scheme - this model is future referred to as Char+Word SVM. Onose et al. (2019) used an approach based on a Recurrent Neural Network (RNN) with word vectors from a pre-trained FastText model (Grave et al., 2018) - this model is referred to as BiGRU, as the best results were obtained when using bidirectional GRU cells. Both base and large versions of RoBERT outperform previous state of the art models for the binary classification task, with RoBERT-large setting the new state of the art. For the MD to RO topic classification task, the best BERT-based model is BERT-base-ro with a marginal 0.02 increase over RoBERT-large, but the BiGRU approach proposed by Onose et al. (2019) obtains the best macro F1 score out of all considered models. For the last task, RoBERT-large outperforms all other models, obtaining a better macro F1 than the previous state of the art.

Model	Dialect Classification	MD to RO	RO to MD
2-CNN + SVM	89.50 (93.40)*	38.56 (65.09)*	44.72 (75.21)*
Char+Word SVM	75.73 (96.20)*	61.15 (69.08)*	55.33 (81.93)*
BiGRU	70.88 (93.30)*	48.13 (**70.10**)*	48.08 (80.30)*
mBERT	95.34	68.76	78.24
XLM-R-base	96.28	69.93	82.28
BERT-base-ro	96.20	69.93	78.79
RoBERT-small	95.67	69.01	80.40
RoBERT-base	97.39	68.30	81.09
RoBERT-large	**97.78**	69.91	**83.65**

* After competition

Table 7: Macro-averaged F1 score (in %) for the test set of VarDial 2019 MRC.

5.4 Diacritics restoration

The diacritics restoration task is framed as a classification problem. The classes were the following: make no modification to the current character (e.g., *a* → *a*), add circumflex mark (e.g., *a* → *â* and *i* → *î*), add breve mark (e.g., *a* → *ă*), and two more classes for adding comma below (e.g., *s* → *ș* and *t* → *ț*). This leads to a total of 5 classes for our classification problem.

A basic model is considered and future improvements are provided. First, a character-level convolutional neural network (CharCNN) as proposed by Kim et al. (2016) was implemented. Specifically, a window of size 11 (meaning 5 characters to the left and 5 characters to the right of the middle character) is considered for each character that can accept a diacritic mark. All 11 characters are passed through an embedding layer of size 50, which results in a matrix E^{11x50}. On top of this matrix, a CNN is used with filter widths of 2, 3, 4, 5, and 11, and 50 filters for each width. A max-over-time pooling is further applied to obtain a fixed representation for the current window; this leads to a vector of 250 elements for each character window. The embedding representation of the current character (the one in the middle of the window) is concatenated to this vector, followed by a fully connected layer, and a final decision layer. Cross-entropy loss is minimized using the Adam optimizer with a learning rate of 1e-5. For this model (CharCNN), three different architectures were experimented: a) concatenating all character representations after the embedding layer followed by a fully connected layer, and b) two different variants based on CNNs with filter widths of [2, 3, 4, 5] and [2, 3, 4, 5, 11]. From our experiments, the best architecture was the one with filter widths of [2, 3, 4, 5, 11]; this variant was used through all following experiments. In addition, the fully connected layer size was set to 128.

The next step was to integrate context information computed by BERT-based models. For each character that can accept a diacritic mark, a BERT-based model (i.e., mBERT, XLM RoBERTa, BERT-base and RoBERT variants) is used to compute the representation of the word (token) that contains the mentioned character. The current sentence is passed through the BERT model and the needed token representation is extracted, in the same way BERT is used for tagging tasks. This semantic representation is concatenated with the character-level representation, further passed to a fully connected layer followed by the decision layer. Two different setups were considered: a) having the BERT model frozen and used as feature extractor, and b) training the entire architecture including BERT. The architecture with BERT-layer frozen is trained for 20 epochs with Adam optimizer (Kingma and Ba, 2015) and a learning rate of 1e-3. The entire architecture training (BERT-layer trainable) continues from the previous best model for a total of 5 epochs, using the same optimizer with a learning rate of 1e-5.

Both word-level and character-level accuracy are used for evaluation, with two different variants: taking into account only words/characters that accept diacritics ($word_{dia}$ and $char_{dia}$), and using all words/characters ($word_{all}$ and $char_{all}$); this leads to four metrics in total.

Iordache et al. (2019) used an approach based on character-level Recurrent Neural Networks (RNNs) and the top performing model uses a two-layer bidirectional LSTM (BiLSTM) with 2.4M trainable parameters. They also consider the task as a classification problem with 3 different classes (no diacritics / î, ș, ț, â / ă) and use cross-entropy as loss function - their approach is further referred to as BiLSTM.

Table 8 presents the results on the diacritics restoration challenge. All variants of RoBERT outperform mBERT across all metrics, on both development and test set. In addition, a rather large gap in performance between base and large version of RoBERT exists, in favor or the former. Most likely, this happens because of the larger token representation space of RoBERT-large, when compared to RoBERT-base (1024 vs 768). This representation is further concatenated with the much smaller CNN output (of size 300) and then passed through a 128-sized fully connected layer. When fine-tuning the entire architecture (including the BERT model), this gap has almost vanished. Lastly, the best model is based on the BERT-base-ro model, with RoBERT-base close by with a marginal 0.01 or 0.02 difference. A larger gap in performance is observed when the BERT layer is frozen; for this reason, we believe that BERT-base-ro has a slight advantage on this task due to its larger vocabulary size (50k versus 38k tokens).

Model	**Performance on dev set**				**Performance on test set**			
	word$_{dia}$	**word**$_{all}$	**char**$_{dia}$	**char**$_{all}$	**word**$_{dia}$	**word**$_{all}$	**char**$_{dia}$	**char**$_{all}$
BiLSTM	-	-	-	-	-	99.42	-	-
CharCNN	98.05	98.41	99.08	99.71	98.03	98.40	99.07	99.65
CharCNN + mBERT*	98.80	99.02	99.44	99.82	98.80	99.02	99.43	99.78
CharCNN + XLM-R-base*	99.46	99.56	99.74	99.92	99.44	99.54	99.73	99.90
CharCNN + BERT-base-ro*	99.64	99.71	99.83	99.95	99.62	99.69	99.82	99.93
CharCNN + RoBERT-small*	99.20	99.35	99.62	99.88	99.19	99.34	99.61	99.85
CharCNN + RoBERT-base*	99.52	99.61	99.77	99.93	99.51	99.60	99.77	99.91
CharCNN + RoBERT-large*	98.86	99.07	99.47	99.83	98.82	99.03	99.44	99.79
CharCNN + mBERT	99.68	99.74	99.85	99.95	99.66	99.72	99.84	99.94
CharCNN + XLM-R-base	99.73	99.78	99.87	**99.96**	99.71	99.76	99.86	**99.95**
CharCNN + BERT-base-ro	**99.75**	**99.80**	**99.88**	**99.96**	**99.74**	**99.79**	**99.88**	**99.95**
CharCNN + RoBERT-small	99.69	99.75	99.85	99.95	99.67	99.73	99.84	99.94
CharCNN + RoBERT-base	99.74	99.79	**99.88**	**99.96**	99.73	99.78	99.87	**99.95**
CharCNN + RoBERT-large	99.73	99.78	99.87	**99.96**	99.70	99.76	99.86	**99.95**

* BERT-layer frozen

Table 8: Performance metrics on diacritics restoration challenge.

6 Conclusions

Three Romanian language models based on BERT architecture were pre-trained, evaluated, and publicly released. We compare our models with strong baselines represented by both multi-lingual models and another pre-trained Romanian language model. Our results indicate that mono-lingual models consistently outperform multi-lingual models, in spite of the fact that the former have fewer parameters, and the latter are pre-trained on significantly more data.

Since our objective was to compare our models with existing multi-lingual ones, we kept a very similar architecture and very little parameter tuning across all tasks. Better results could probably be obtained for a specific task by experimenting with additional hyper-parameters. Nevertheless, our models set the state of the art results on almost all considered tasks.

The release of pre-trained BERT models is an important step for NLP progress in languages with limited resources, such as Romanian. Overall, pre-trained models are easily incorporated, introduce important performance improvements, and help researchers tackle new problems, when task-specific datasets are scarce, because fine-tuning a model requires fewer examples in contrast to a model trained from scratch.

Acknowledgments

This research was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-III 54PCCDI /2018, INTELLIT – "Prezervarea și valorificarea patrimoniului literar românesc folosind soluții digitale inteligente pentru extragerea și sistematizarea de cunoștințe", by the "Semantic Media Analytics – SeMAntic" subsidiary contract no. 20176/30.10.2019, from the NETIO project ID: P_40_270, MySMIS Code: 105976, as well as by "Spacetime Vision – Towards Unsupervised Learning in the 4D World", project Code: EEA-RO-NO-2018-0496.

References

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

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*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam Mccandlish, Alec Radford, Ilya Sutskever, and Dario Amodei Openai. 2020. Language Models are Few-Shot Learners. *CoRR*, abs/2005.1.

Andrei M Butnaru and Radu Tudor Ionescu. 2019. MOROCO: The Moldavian and Romanian Dialectal Corpus. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 688–698.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, and Jorge Pérez. 2020. Spanish Pre-Trained BERT Model and Evaluation Data. In *to appear in PML4DC at ICLR 2020*.

Junyoung Chung, Caglar Gulcehre, and Kyunghyun Cho. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR*, abs/1412.3.

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, ACL 2020*, pages 8440–8451. Association for Computational Linguistics.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. *CoRR*, abs/1912.0.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. *CoRR*, abs/2001.0.

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, NAACL-HLT 2019*, pages 4171–4186. Association for Computational Linguistics.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Florin Iordache, Lucian Georgescu, Dan Oneață, and Horia Cucu. 2019. Romanian Automatic Diacritics Restoration Challenge. In *14th Edition of the International Conference on Linguistic Resources and Tools for Natural Language Processing (CONSILR), Cluj-Napoca, Romania*.

Pedro Javier Ortiz Suárez, Benoît Sagot, Laurent Romary, and Beno^ıtBeno^ıt Sagot. 2019. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-Aware Neural Language Models. In *AAAI'16: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*.

Diederik Kingma and Jimmy Ba. 2015. Adam: a method for stochastic optimization. *3rd International Conference for Learning Representations*.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. *Komp'juternaja Lingvistika i Intellektual'nye Tehnologii*, 2019-May(18):333–339.

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*.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. FlauBERT: Unsupervised Language Model Pretraining for French. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020*, pages 2479–2490. European Language Resources Association.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.1.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte 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, ACL 2020*, pages 7203–7219. Association for Computational Linguistics.

Shikib Mehri and Giuseppe Carenini. 2017. Chat Disentanglement : Identifying Semantic Reply Relationships with Random Forests and Recurrent Neural Networks. In *IJCNLP2017*, number 2003, pages 615–623.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *Advances in neural information processing systems*, pages 1–9.

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. PhoBERT: Pre-trained language models for Vietnamese. *CoRR*, abs/2003.0.

Cristian Onose, Dumitru-Clementin Cercel, and Stefan Trausan-Matu. 2019. SC-UPB at the VarDial 2019 Evaluation Campaign: Moldavian vs. Romanian Cross-Dialect Topic Identification. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 172–177.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, and Valerio Basile. 2019. ALBERTO: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets. In *Proceedings of the Sixth Italian Conference on Computational Linguistics*. CEUR-WS.org.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. Technical report, OpenAI.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. Technical report, OpenAI.

Samuel Rönnqvist, Jenna Kanerva, Tapio Salakoski, and Filip Ginter. 2019. Is Multilingual BERT Fluent in Language Generation? *CoRR*, abs/1910.0.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2019. Portuguese Named Entity Recognition using BERT-CRF. *CoRR*, abs/1909.1.

Diana-Elena Tudoreanu. 2019. DTeam @ VarDial 2019: Ensemble based on skip-gram and triplet loss neural networks for Moldavian vs. Romanian cross-dialect topic identification. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 202–208.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in neural information processing systems*, pages 5998–6008.

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. *CoRR*, abs/1912.0.

Shuohang Wang and Jing Jiang. 2017. Machine Comprehension Using Match-LSTM and Answer Pointer. In *5th International Conference on Learning Representations, ICLR 2017*.

Lidan Wang and Ming Tan. 2016. FastHybrid : A Hybrid Model for Efficient Answer Selection. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING-16)*, pages 2378–2388.

Nianheng Wu and Eric Demattos Kwok. 2019. Language Discrimination and Transfer Learning for Similar Languages: Experiments with Feature Combinations and Adaptation. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 54–63.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XL-Net: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS 2019*, pages 5754–5764.