Exploring the Power of Romanian BERT for Dialect Identification

George-Eduard Zaharia[1]*, **Andrei-Marius Avram**[1, 2]*,
Dumitru-Clementin Cercel[1], **Traian Rebedea**[1]
University Politehnica of Bucharest, Faculty of Automatic Control and Computers[1]
Research Institute for Artificial Intelligence, Romanian Academy[2]
{george.zaharia0806, andrei_marius.avram}@stud.acs.upb.ro
{dumitru.cercel, traian.rebedea}@upb.ro

Abstract

Dialect identification represents a key aspect for improving a series of tasks, such as opinion mining, considering that the location of the speaker can greatly influence the attitude towards a subject. In this work, we describe the systems developed by our team for VarDial 2020: Romanian Dialect Identification, a task specifically created for challenging participants to solve the dialect identification problem for an under-resourced language, such as Romanian. More specifically, we introduce a series of neural architectures based on Transformers, that combine a BERT model exclusively pre-trained on the Romanian language with several other techniques, such as adversarial training or character-level embeddings. By using a custom Romanian BERT model, we were able to reach a macro-F1 score of 64.75 on the test dataset, thus allowing us to be ranked 5^{th} out of 8 participant teams. Moreover, we improved the F1-scores reported by the authors of MOROCO with over 1.7%, obtaining a 96.23% macro-F1 score, alongside micro and weighted F1 scores of 96.25%.

1 Introduction

Currently, the Romanian language is still considered an under-resourced language, although in the recent years, several datasets were created that tried to mitigate this problem such as the reference corpus of the Contemporary Romanian Language (CoRoLa) (Mititelu et al., 2018), the Romanian Named Entity Corpus (RONEC) (Dumitrescu and Avram, 2019), the Biomedical Gold Standard Corpus (MoNERo) (Mitrofan et al., 2019), the Romanian Speech Corpus (RSC) (Georgescu et al., 2020), and the Romanian WordNet (Dumitrescu et al., 2018). With the rise of attention-based language models, the first Romanian Bidirectional Encoder Representations from Transformer (Ro-BERT) appeared and it outperformed Multilingual BERT (M-BERT) (Pires et al., 2019) on all the evaluation tasks (Dumitrescu et al., 2020)[1].

One of the most addressed and challenging tasks in natural language processing research is text dialect identification. As a response to this challenge, Butnaru and Ionescu (2019) introduced the Moldavian and Romanian Dialectal Corpus (MOROCO), a dataset that contains 33,564 samples of text collected from news websites, grouped in two dialects using the top level domain of the websites: Romanian and Moldavian (".ro" and ".md"). Moreover, a shared task, called Romanian Dialect Identification (RDI), was proposed at VarDial 2020 (Găman et al., 2020) and it aimed to evaluate the performance of each participant system on this corpus.

Starting from the MOROCO dataset, the RDI competition introduces the challenge of properly identifying the Romanian or Moldavian dialect, considering that the test dataset is from a different domain. That is, the validation dataset contains long texts, written in either the Romanian or the Moldavian dialect, while the test dataset is composed of short entries, based on tweets. Therefore, this difference

*These authors contributed equally.

[1] Romanian BERT comes in the cased and uncased variants, that are open-sourced at:
https://github.com/dumitrescustefan/Romanian-Transformers

Proceedings of the 7th VarDial Workshop on NLP for Similar Languages, Varieties and Dialects, pages 232–241
Barcelona, Spain (Online), December 13, 2020

influenced the performance of our models, considering that we were able to obtain a 97.04% macro-F1 score on the validation set, while the evaluation on the test set yielded a 64.75% macro-F1 score.

This work is structured as follows. In Section 2, we perform an analysis of existing solutions for closely related dialect identification tasks. Section 3 outlines our solutions for the dialect identification issue, while Section 4 details the performed experiments, experimental setup, and error analysis. Finally, we draw conclusions in Section 5.

2 Related Work

There are various approaches regarding the language dialect identification task. Some of them are centered around the Romanian language, while others are focused on different ones, such as the Arabic or German dialects. However, they are equally important, considering that some techniques can cross the language barrier and be used as universal dialect identification methods.

2.1 Romanian Dialect Identification

For example, previous work (Onose et al., 2019) in Romanian dialect identification employed the usage of various deep learning models, including Recurrent Neural Networks (RNNs) (Elaraby and Abdul-Mageed, 2018a), Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), and Gated Recurrent Units (GRUs) (Cho et al., 2014), alongside various word embeddings. Furthermore, Tudoreanu (2019) applied an ensemble of neural networks that uses a triplet loss alongside Convolutional Neural Networks (CNNs) (Kim, 2014), with the purpose of maximizing the distance between an anchor sample and a negative example while minimizing the difference between the anchor and the positive example. Wu et al. (2019) also considered Support Vector Machines (Cortes and Vapnik, 1995), but paired with character n-grams.

2.2 Dialect Identification for Other Languages

Aiming to tackle the Arab dialect identification problem by participating at the MADAR shared task (Bouamor et al., 2019), Abdul-Mageed et al. (2019) introduced a series of solutions based on traditional, deep learning, Natural Language Processing (NLP) techniques, like GRUs and, at the same time, state-of-the-art, Transformer-based methods, i.e., BERT (Zhang and Abdul-Mageed, 2019). Moreover, Salameh et al. (2018) engaged in the same problem by employing a solution based on features, including character and word n-grams and applying a Multinomial Naive Bayes classifier. Similar traditional methods were also applied by Elaraby and Abdul-Mageed (2018b) using logistic regression, SVMs, and, moreover, models based on RNNs. Other work (Butnaru and Ionescu, 2018) proposed string kernel functions (Lodhi et al., 2002) that capture the similarity between text samples based on character n-grams, while a different approach (Ali, 2018) simply implies the usage of CNNs.

Employing similar techniques, but switching the language, Malmasi and Zampieri (2017) addressed the German dialect identification issue by also using traditional machine learning techniques, but, furthermore, adding different ensemble classifiers. Further focusing on the German language, Gaman and Ionescu (2020) proposed several methods for approaching the previously mentioned subject, including character-level CNNs, Support Vector Regressors based on string kernels and ensemble learning systems (Chen and Guestrin, 2016).

3 Methods

We focused our approaches around Transformers (Vaswani et al., 2017), considering that they represent state-of-the-art solutions for solving NLP problems.

3.1 Vanilla Transformer-based Solutions

3.1.1 Multilingual BERT

Aimed for multilingual NLP problems, M-BERT is a variant of BERT (Devlin et al., 2018), pre-trained on over 100 languages, thus ensuring good performance for all of them, not only for the English language.

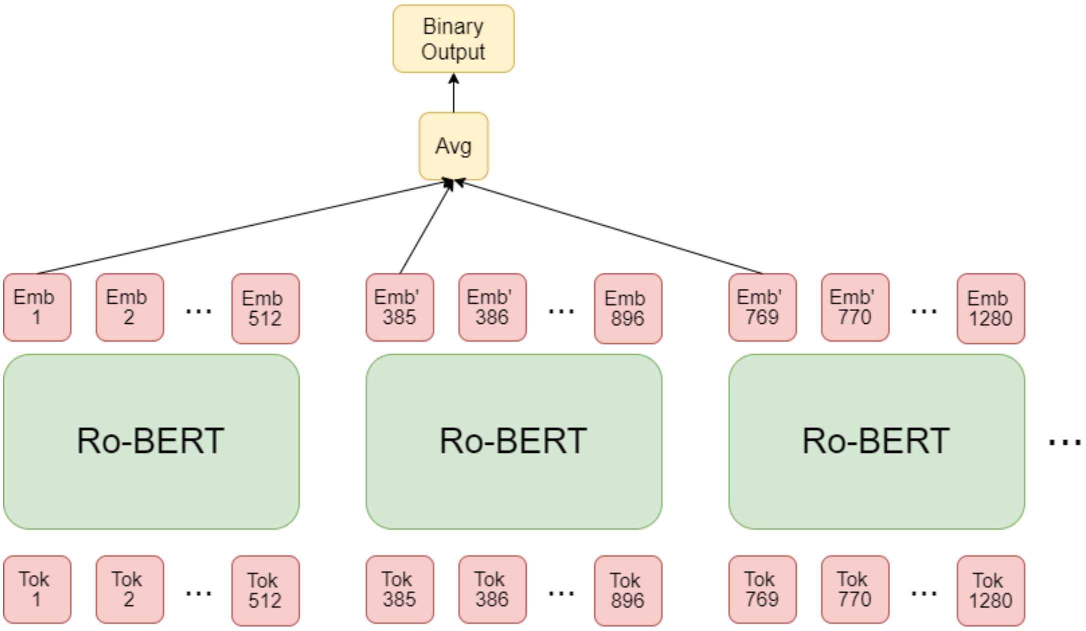

Figure 1: The architecture of Custom-Ro-BERT-FT. The *Emb'* notation shows that the respective embedding is different from the embedding obtained on the same tokens, on the previous sequence.

M-BERT can be used for a wide array of tasks, including sequence classification, therefore allowing us to fine-tune the model for our problem, dialect identification in Romanian.

3.1.2 Romanian BERT

We also experimented with the embeddings obtained from the Ro-BERT model. Ro-BERT was trained on three publicly available corpora: OPUS (Tiedemann, 2012), OSCAR (Suárez et al., 2019), and Wikipedia, using masked language modeling and next sentence prediction as training objectives. To validate the resulted model, the performance of Ro-BERT was compared with the performance of M-BERT on three tasks from Romanian corpora: (1) Simple Universal Dependencies - the models had to predict independently the Universal Part-of-Speech (UPOS) and the eXtended Part-of-Speech, (2) Joint Universal Dependencies - the models had to jointly predict the UPOS, Universal Features, Lemmas and Dependency Parsing, and (3) Named Entity Recognition - the models had to predict the BIO labels. For the first two tasks, the authors used the Romanian RRT corpus (Barbu Mititelu et al., 2016), while for the last one RONEC (Dumitrescu and Avram, 2019). The evaluation results showed that Ro-BERT outperformed M-BERT on all tasks with values ranging between 1% and 3%.

3.2 Proposed Approaches

3.2.1 Custom Ro-BERT Fine-tuning

To use the Transformer-based language models on the competition dataset, we firstly tokenized the sentences by using the Byte-Pair Encoding (BPE) tokenizer with the additional NE token. As depicted in Table 1, some of the sequences can be very long, so applying the model directly on them as described in Devlin et al. (2018) is not optimal. To mitigate this problem, we applied the model on consecutive sequences of 512 tokens that share the first 128 tokens with the previous sequence. Then, to create a binary output, we averaged the embeddings of all tokens out of each 512 token sequence and projected it into a scalar. This process is further depicted in Figure 1. We will further reference this system under the name of Custom-Ro-BERT-FT.

3.2.2 Embedding Concatenation

Next, we intended to enhance the word representations with information at the morpheme-level, therefore, we needed to use character-level embeddings. By breaking each word into a sequence of characters,

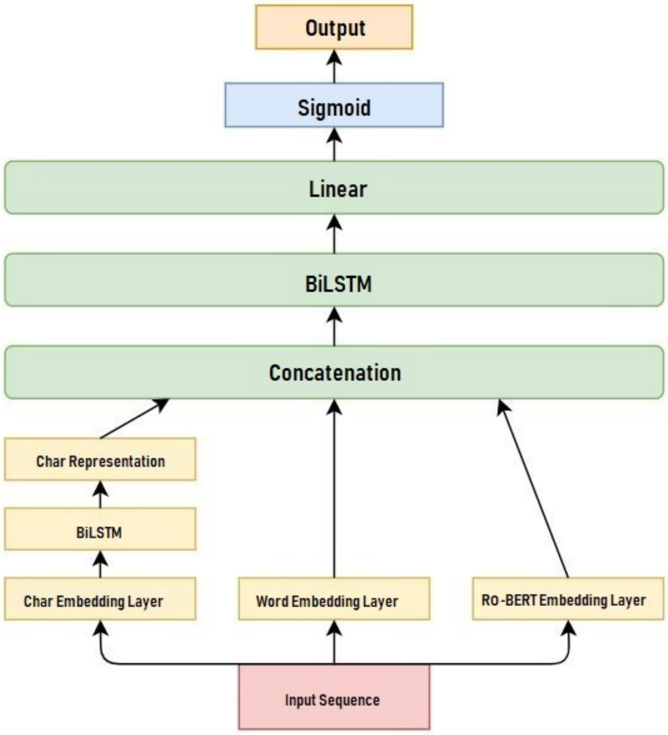

Figure 2: The architecture of the EC model.

then mapping them to a series of indexes and then feeding them into a Bidirectional LSTM (BiLSTM), we were able to obtain another set of representations for the inputs. The character-level embeddings allowed us to identify structural similarities between different words, an important aspect when tackling a dialect identification problem.

Furthermore, we also added pre-trained fastText word embeddings (Bojanowski et al., 2017). The three representations (i.e., Transformer embeddings, word embeddings, and character embeddings) were concatenated, making sure that the first dimension is identical for all of them, representing the number of input tokens. The resulted tensor was then fed to a BiLSTM network and then to a linear layer, thus obtaining the final representation for the input sequence. Finally, we used the sigmoid activation function for obtaining the final class. Figure 2 depicts the previously described architecture, called EC.

3.2.3 Adversarial Training

Initially applied for improving computer vision solutions, adversarial training (Goodfellow et al., 2014) represents a technique that intentionally alters a percentage of training entries with perturbations. Even though the changes are minimal, the effects on the performance of the system can be major, since the perturbations can lead to missclassifications. The previously mentioned process can be applied to both text and image models. Because of the generalization it creates, it can lead to improved performance for the first category. Therefore, language models achieve better results when trained under this approach.

Since we intended to also maximize the performance of our models, we resorted to an adversarial training technique. Therefore, we used FreeLB (Zhu et al., 2019), an enhanced adversarial training method for natural language models. FreeLB performs adversarial training by introducing adversarial perturbations at word-level embeddings and then minimizing the adversarial loss resulted from the input samples. The model receives training data in batches that are affected by the adversarial algorithm, namely, they are augmented with extra adversarial entries. Each iteration creates some outputs, the purpose of the FreeLB algorithm being to take the gradients of these outputs and to average them. Furthermore, FreeLB minimizes the maximum risk at each ascent step, with the advantage of creating an insignificant overhead.

4 Experiments

4.1 Dataset Analysis

	White Space Tokenizer		Ro-BERT Tokenizer		M-BERT Tokenizer	
Dataset	**Avg. Tokens**	**Max. Tokens**	**Avg. Tokens**	**Max. Tokens**	**Avg. Tokens**	**Max. Tokens**
Train (MOROCO)	310.04	15988	356.08	18456	449.70	24169
Valid (MOROCO)	309.92	10809	355.87	12578	450.02	16676
Test (MOROCO)	313.65	13213	360.87	15313	455.50	20151
Test (RDI)	15.63	25	21.71	42	26.65	44

Table 1: Statistics of the datasets we used in our experiments, MOROCO-RDI and RDI.

MOROCO was created by collecting texts from the top news websites in Romania and Moldavia, and by automatically labeling them using the Internet domain, resulting in 33,564 samples (45.89% Moldavian and 54.11% Romanian) having a total of more than 10 million tokens. The news were selected from six domains: culture, finance, politics, science, sports, and tech. The authors further processed the text by removing all HTML tags and by replacing the named entities with the NE token in order to prevent the models from classifying based on features that are not specific to the dialect, but to the environment in which the dialect is used. Moreover, in order to provide a proper comparison with other similar corpora, five tasks were created on the MOROCO data set: binary classification by dialect (MOROCO-RDI), intra-dialect classification by topic using the Romanian or the Moldavian samples, and cross-dialect topic classification by training a model on the samples of one dialect and testing on the other dialect set of samples. The dataset was also split into training, validation and testing, resulting in subsets that contained 21,719, 5,921 and 5,924 number of samples.

At the evaluation phase of the RDI task, a new data set was used to evaluate the performance of the submitted models. The new set contained 5,022 samples, mostly taken from social media.

Further, we analyzed the two datasets (MOROCO and RDI) in Table 1 by computing the average number of tokens and the maximum number of tokens, using the white space tokenizer, the Romanian BERT uncased tokenizer, and the M-BERT uncased tokenizer. We note that the change in domain led to a significant difference in the number of tokens, of several orders of magnitude, which in turn made our models to perform much worse on the RDI test set than we initially estimated on the MOROCO test set.

4.2 Implementation Details

For the EC solution, we considered the Adam optimizer (Kingma and Ba, 2014) with a 0.001 learning rate. Furthermore, the BiLSTM hidden size is 500, while the input maximum length is 280 tokens. We trained the model for 8 epochs, by using an early stopping policy. At the same time, for the adversarial training method, we used an initial learning rate of 5e-5, alongside the Adam optimizer. Moreover, the weight decay and the epsilon parameters were kept with their default values 0.0 and 1e-8 respectively, and the training process spanned over 12 epochs. For the standard Ro-BERT and also for the custom Ro-BERT fine-tuning process, we employed the Adam with weight decay (AdamW) optimizer (Loshchilov and Hutter, 2017) with a 2e-5 learning rate, for 4 epochs.

4.3 Custom Language Model Comparison

The first experiment we conducted was a comparison between M-BERT and Ro-BERT on the MOROCO-RDI test dataset in order to choose a language model to work with. At this stage, because the entries had a high number of tokens, we also experimented with various N, i.e., the number of consecutive sequences with 512 tokens that share 128 tokens with the previous sequence, on which the language model is applied. The maximum number of tokens (determined by N) and by using the Ro-BERT and M-BERT tokenizers is presented in Table 2, together with the percentage of samples that have fewer tokens than the maximum. Also, the results are depicted in Figure 3. The left figure presents the case where all samples that have more tokens than the maximum allowed, are dropped both from the train set

No. of Apply (N)	No. of Tokens	M-BERT Perc.	Ro-BERT Perc.
1	512	72.38%	83.60%
2	896	92.28%	95.50%
3	1280	96.62%	98.04%
4	1536	97.78%	98.64%

Table 2: Maximum number of tokens allowed for a given N and the percentage of samples that satisfy this condition.

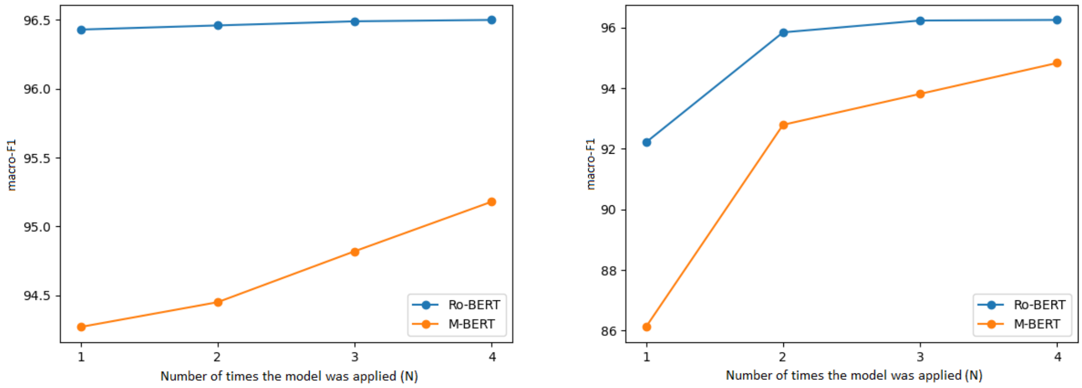

Figure 3: M-BERT and Ro-BERT comparison with various N - the number of times each language model is applied, trained on the MOROCO-RDI dataset and tested on samples from the test set that do not have a token length longer than the maximum allowed in the training set (left) or on the whole test set (right).

and the test set, while the right figure keeps all the samples from the test set, but also drops the samples from the training set that do not meet the requirements. It can be observed that in both cases Ro-BERT offers a better performance for all values of N and that the rate of change in performance of M-BERT improves faster than the performance of Ro-BERT, maybe even surpassing Ro-BERT for a large number of tokens[2].

At evaluation time, for the RDI dataset, the sequences were much smaller than the ones from the initial dataset (MOROCO-RDI), and in order to avoid overfitting on long sequences, we used the system that applies Ro-BERT only 3 times instead of 4 times. Moreover, our choice is further motivated by the fact that the difference in performance between $N = 3$ and $N = 4$ is rather small (0.6%).

4.4 Submitted Results

Next, we experimented with four different types of architectures, out of which we used the best three for our VarDial submissions. Table 3 presents the results obtained on the entire test datasets. The MOROCO-RDI dataset contains entries similar to the ones used for training and validation. On the other hand, the RDI shared task counterpart contains much shorter entries, obtained from tweets.

The best results obtained for the RDI dataset are yielded by using the Custom-Ro-BERT-FT technique, with a weighted-F1 score of 64.80%, alongside a 64.75% micro-F1 and a 67.10% macro-F1. The closest result further obtained by our experiments comes at a difference of 8.41% in terms of weighted-F1 score, provided by the FreeLB technique, with a value of 56.39%. Also, the same experiment produced a 60.77% micro-F1 score and a 56.32% macro-F1 score. Furthermore, the EC solution proves to offer poorer results, considering the increased width and depth of the neural network and thus the large number of parameters that needed to be fine-tuned. The main metric, the weighted-F1 score, has a value of 46.59%, while the others, the macro and micro F1 measures, have values of 46.48% and 55.07%,

[2] Unfortunately, we could not make this analysis due to the lack of computational resources.

Task	Method	Micro-F1	Weighted-F1	Macro-F1
RDI	Standard Ro-BERT Fine-tuning	60.77	55.82	55.74
	Custom-Ro-BERT-FT	**67.10**	**64.80**	**64.75**
	FreeLB	60.77	56.39	56.32
	EC	55.07	46.59	46.48
	Top Model	78.75	78.76	78.77
MOROCO-RDI	Standard BERT Fine-tuning	94.48	94.52	94.51
	Custom-Ro-BERT-FT	**96.25**	**96.25**	**96.23**
	FreeLB	96.15	96.15	96.12
	EC	86.17	86.43	86.31
	Butnaru and Ionescu (2019)	94.60	94.11	94.50

Table 3: Results obtained by our models on the test sets.

respectively.

If we focus our attention on the MOROCO-RDI test dataset format, we can see that the performance difference is considerable. With a 96.25% weighted-F1 score and very close values for macro and micro F1, the Custom-Ro-BERT-FT technique offers the best results, closely followed by FreeLB at a margin of 0.1% in weighted-F1 score, with a value of 96.15%. Moreover, the micro and macro F1 scores have values of 96.15% and 96.12%, respectively. The EC model offers a value of 86.43% weighted-F1 alongside 86.17% and 86.31% micro and macro F1 scores. Furthermore, the standard Ro-BERT fine-tuning comes second to last in terms of performance, with a 1.73% difference in weighted-F1 score when compared to the Custom-Ro-BERT-FT model.

4.5 Error Analysis

Table 4 presents examples of entries correctly or wrongly classified. As seen, most of the incorrect entries are part of the Moldavian dialect. The main reason behind the misclassifications is represented by the domain and length differences between the train and development datasets and the evaluation dataset. For training and validation, the average number of tokens is about 310, while for the evaluation dataset, the number is around 15. This discrepancy does not allow the models to properly detect the dialect form the test entries, considering that, in some situations, there are no proper key features that can point towards a Romanian or a Moldavian dialect. For example, the last two examples of misclassified entries from Table 4 do not show any defining aspects of either one of the dialects. Moreover, the fourth one contains only one proper word, ”Primele” (eng. ”The first”), while the other words are masked by the NE token. This may be an important problem for the task and dataset at hand, as the differences between Moldavian and Romanian are minor and might not arise in short fragments of text such as tweets. For future versions, these datasets should be manually curated to contain more relevant samples.

On the other hand, some entries have clear indicators that point towards a certain dialect. As an example, the root word ”raion” (eng. ”district”), specific to the Moldavian dialect, is a clear indicator of the origin of that input. Additionally, some named entities are not masked and are also present in the training dataset, thus clearing the origin of the including text (e.g., the named entities ”Dodon” or ”Chicu” in the first two correctly classified samples).

5 Conclusion and Future Work

This paper presented our approaches regarding the Romanian Dialect Identification task, organized by VarDial 2020. We proposed a series of Transformer-based architectures that intended to solve the dialect identification issue. All the solutions employ the usage of Ro-BERT, a Transformer model pre-trained on Romanian language corpora. By fine-tuning Ro-BERT with two different techniques, standard and custom, we were able to achieve good scores on both the MOROCO-RDI test dataset and the RDI dataset, used for this year’s competition. Also, by using an adversarial training technique (FreeLB) on Ro-BERT, we improved the state-of-the-art score on the MOROCO-RDI dataset, while the performance

Category	Entry	True label
Correct	1) Dodon: $NE$avut în trecut un guvern, un stat capturat, a urmat un $NE$ condus de $NE$ Dodon: In the past, $NE$ had a government, a captured state, followed by a $NE$ lead by $NE$	MD
	2) Chicu crede crearea platformelor industriale în fiecare centru raional o soluție de renaștere a economiei naționale Chicu believes that the creation of industrial platforms in each district center represents a solution for the rebirth of the national economy	MD
	3) Seri de tango și saloane de flori la un spital de psihiatrie din $NE$ Un psiholog aduce speranță unor oameni ca Tango evenings and flower salons at a psychiatric hospital in $NE$ A psychologist brings hope to people like	RO
Wrong	1) Pericol pentru $NE$ $NE$ vor revenirea a $NE$ $NE$ de militari ruși în zona de securitate Danger to $NE$ $NE$ want the return of $NE$ $NE$ Russian military in the security zone	MD
	2) FOTO $NE$ $NE$ $NE$ iarna a devenit primăvară. Un arbust ornamental a înflorit, $NE$ de vremea caldă PHOTO $NE$ $NE$ $NE$ winter has become spring. An ornamental shrub bloomed, $NE$ because of the warm weather	RO
	3) Cum te protejezi împotriva coronavirusului $NE$ How to protect yourself against the coronavirus $NE$	MD
	4) Primele $NE$ $NE$ $NE$ $NE$ $NE$ $NE$ The first $NE$ $NE$ $NE$ $NE$ $NE$ $NE$	MD

Table 4: Examples of correctly and wrongly classified entries. MD: Moldavian, RO: Romanian.

decreased on the RDI set. Moreover, employing an embedding concatenation technique does not help with performance, yielding the poorest results among the four techniques we experimented with.

For future work, we intend to also experiment with multi-task learning approaches (Caruana, 1997), considering that, usually, an auxiliary task can help the model to detect additional features that can lead to increased performance. Another aspect we plan to test are CoRoLA-based word embeddings, which can replace their counterpart in the embedding concatenation experiment.

References

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Arun Rajendran, and Lyle Ungar. 2019. Dianet: Bert and hierarchical attention multi-task learning of fine-grained dialect. *arXiv preprint arXiv:1910.14243*.

Mohamed Ali. 2018. Character level convolutional neural network for arabic dialect identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 122–127.

V Barbu Mititelu, Radu Ion, Radu Simionescu, Elana Irimia, and Cenel-Augusto Perez. 2016. The romanian treebank annotated according to universal dependencies. In *Proceedings of the tenth international conference on natural language processing (hrtal2016)*.

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.

Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The madar shared task on arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 199–207.

Andrei M Butnaru and Radu Tudor Ionescu. 2018. Unibuckernel reloaded: First place in arabic dialect identification for the second year in a row. *arXiv preprint arXiv:1805.04876*.

Andrei Butnaru and Radu Tudor Ionescu. 2019. Moroco: The moldavian and romanian dialectal corpus. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 688–698.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Corinna Cortes and Vladimir Vapnik. 1995. Support vector machine. *Machine learning*, 20(3):273–297.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Stefan Daniel Dumitrescu and Andrei-Marius Avram. 2019. Introducing ronec–the romanian named entity corpus. *arXiv preprint arXiv:1909.01247*.

Stefan Daniel Dumitrescu, Andrei Marius Avram, Luciana Morogan, and Stefan-Adrian Toma. 2018. Rowordnet–a python api for the romanian wordnet. In *2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–6. IEEE.

Stefan Daniel Dumitrescu, Andrei-Marius Avram, and Sampo Pyysalo. 2020. The birth of romanian bert. *arXiv preprint arXiv:2009.08712*.

Mohamed Elaraby and Muhammad Abdul-Mageed. 2018a. Deep models for arabic dialect identification on benchmarked data. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 263–274.

Mohamed Elaraby and Muhammad Abdul-Mageed. 2018b. Deep models for arabic dialect identification on benchmarked data. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 263–274.

Mihaela Gaman and Radu Tudor Ionescu. 2020. Combining deep learning and string kernels for the localization of swiss german tweets. *arXiv preprint arXiv:2010.03614*.

Alexandru-Lucian Georgescu, Horia Cucu, Andi Buzo, and Corneliu Burileanu. 2020. Rsc: A romanian read speech corpus for automatic speech recognition. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6606–6612.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Mihaela Găman, Dirk Hovy, Radu Tudor Ionescu, Heidi Jauhiainen, Tommi Jauhiainen, Krister Lindén, Nikola Ljubešić, Niko Partanen, Christoph Purschke, Yves Scherrer, and Marcos Zampieri. 2020. A Report on the VarDial Evaluation Campaign 2020. In *Proceedings of the Seventh Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*.

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

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

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

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444.

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

Shervin Malmasi and Marcos Zampieri. 2017. German dialect identification in interview transcriptions. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 164–169.

Verginica Barbu Mititelu, Dan Tufiş, and Elena Irimia. 2018. The reference corpus of the contemporary romanian language (corola). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Maria Mitrofan, Verginica Barbu Mititelu, and Grigorina Mitrofan. 2019. Monero: a biomedical gold standard corpus for the romanian language. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 71–79.

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.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. Fine-grained arabic dialect identification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 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)*. Leibniz-Institut für Deutsche Sprache.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218.

Diana 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, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Nianheng Wu, Eric DeMattos, Kwok Him So, Pin-zhen Chen, and Çağrı Çöltekin. 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.

Chiyu Zhang and Muhammad Abdul-Mageed. 2019. No army, no navy: Bert semi-supervised learning of arabic dialects. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 279–284.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*.