

The birth of Romanian BERT

Stefan Daniel Dumitrescu **Andrei-Marius Avram** **Sampo Pyysalo**
Independent Researcher Politehnica Univ. of Bucharest University of Turku
Bucharest, Romania Bucharest, Romania Turku, Finland
{dumitrescu.stefan, avram.andreimarius, sampo.pyysalo}@gmail.com

Abstract

Large-scale pretrained language models have become ubiquitous in Natural Language Processing. However, most of these models are available either in high-resource languages, in particular English, or as multilingual models that compromise performance on individual languages for coverage. This paper introduces Romanian BERT, the first purely Romanian transformer-based language model, pretrained on a large text corpus. We discuss corpus composition and cleaning, the model training process, as well as an extensive evaluation of the model on various Romanian datasets. We open source not only the model itself, but also a repository that contains information on how to obtain the corpus, fine-tune and use this model in production (with practical examples), and how to fully replicate the evaluation process.

1 Introduction

A revolution started in natural language processing (NLP) a few years ago, when the first Transformer-based model (Vaswani et al., 2017) demonstrated a significant increase in state-of-the-art results compared to previous neural approaches. The bidirectional BERT (Devlin et al., 2018) language model has since been widely adopted as the baseline for transformer models, and it has been successfully applied to a broad range of NLP tasks from standard language modeling to question answering, text summarization, and machine translation.

A number of papers have been dedicated to studying why and how this model performs so well, including comparison to classical NLP (Tenney et al., 2019), investigation of the newly introduced multi-head attention mechanism (Michel et al., 2019), and analyses of what BERT learns (Clark et al., 2019). A good recent review is presented by Rogers et al. (2020). Following BERT, a number of variations of language models using

Corpus	Lines	Words	Size
OPUS	55.1 M	635.0 M	3.8 GB
OSCAR	33.6 M	1725.8 M	11 GB
Wikipedia	1.5 M	60.5M	0.4 GB
Total	90.2 M	2421.3 M	15.2 GB

Table 1: Corpus statistics (post-cleaning).

the transformer architecture have been introduced, including extended models such as XLM (Lample and Conneau, 2019) and XLNet (Yang et al., 2019) as well as more efficient ones like DistillBERT (Sanh et al., 2019), ALBERT (Lan et al., 2019), and ELECTRA (Clark et al., 2020).

However, the vast majority of these studies have focused only on English models. While Google has released a multilingual BERT model trained on 100+ languages, only recently have monolingual models for other languages started to appear: FlauBERT for French (Le et al., 2019), BERTje for Dutch (de Vries et al., 2019) and FinBERT for Finnish (Virtanen et al., 2019). But none for Romanian, until now. While the multilingual BERT can be used also for Romanian, a monolingual model can bring an increase in accuracy that can be observed also in downstream task performance. Thus, we here introduce `Romanian BERT`. This paper focuses on the technical and practical aspects of `Romanian BERT`, covering corpus composition and cleaning in Section 2, the training process in Section 3, and evaluation on Romanian data sets in Section 4.

2 Corpus

The unannotated texts used to pre-train `Romanian BERT` are drawn from three publicly-available corpora: OPUS, OSCAR and Wikipedia.

OPUS OPUS (Tiedemann, 2012) is a collection of translated texts from the web. It is an open-

Model	Tokenized Sentence
M-BERT (uncased)	cinci bici ##cl ##isti au pl ##eca ##t din cr ##ai ##ova spre so ##par ##lita .
M-BERT (cased)	Ci ##nci bi ##ci ##cl ##i ##ști au pl ##eca ##t din C ##rai ##ova spre Ș ##op ##âr ##li ##ța .
Romanian BERT (uncased)	cinci bicicliști au plecat din craiova spre șopâr ##lita .
Romanian BERT (cased)	Cinci bicicliști au plecat din Craiova spre Șo ##p ##âr ##lita .

Table 2: Model tokenization examples. Note that M-BERT uncased strips accents.

source parallel corpus that was gathered automatically, without human curation. It contains diverse domains of text, from medical prescriptions to law articles and movie subtitles. In total, the OPUS corpus contains around 4GB of Romanian text.

OSCAR OSCAR (Ortiz Suárez et al., 2019), or Open Super-large Crawled ALMANACH coRpus is a huge multilingual corpus obtained by language classification and filtering of the Common Crawl corpus. The Romanian section has about 11GB of text. It contains de-duplicated shuffled sentences.

Wikipedia The Romanian Wikipedia is publicly available for download. We used the February 2020 Wikipedia dump, which contained approx. 0.4GB of text after cleaning.

All corpora were subjected to the same cleaning procedure, with Wikipedia also needing extra cleaning as the XML extraction still contained markup tokens. The cleaning involved several sequential steps:

- Remove all lines under a minimum length.
- Remove all non-UTF8 tokens and all lines that contain forbidden characters / too many numbers (over 25%) / non-ASCII characters (over 40%) / too few letters (under 50%).
- Correct badly hyphenated words (e.g. "a- mi" to "a-mi"), badly hyphenated measure units (e.g. "km/ h" to "km/h"), badly formatted numbers (e.g. "12, 5%" to "12,5%").
- Remove soft hyphens, URLs, emails.
- Normalize dashes (there are several types of Unicode dashes) and other characters.
- Reduce multiple spaces.

3 Pretraining

The pretraining process starts with building a vocabulary on the available corpus. Using byte-pair-encoding (BPE), we generated cased and uncased vocabularies containing 50000 word pieces. Character coverage¹ was set to 2000.

¹To reduce UNKS we use a larger character set to cover less frequent chars/symbols

Vocabulary	Tokens/ Word	UNK/ Word
M-BERT (uncased)	2.00	0.005
M-BERT (cased)	1.82	0.004
Romanian BERT (uncased)	1.41	0.0003
Romanian BERT (cased)	1.37	0.0002

Table 3: Vocabulary statistics.

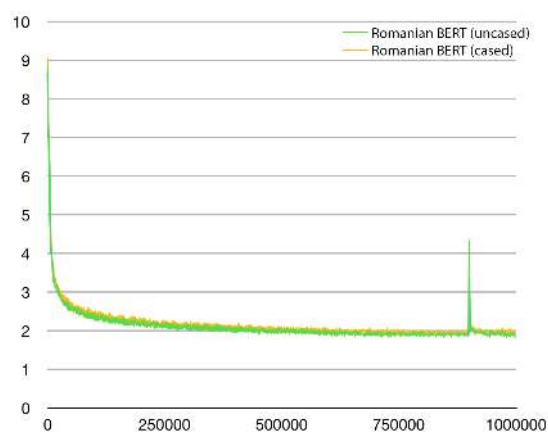


Figure 1: Training loss for the Romanian BERT cased and uncased models.

Vocabulary plays an important part in the performance of a language model. Broadly speaking, the better sentences are tokenized (roughly, the fewer pieces each word is broken into), the better the model is expected to perform. Comparing M-BERT’s vocabulary with ours (Table 3), we see that on average, Romanian BERT is able to encode a word in ~ 1.4 tokens while M-BERT can reach up to 2 tokens/word for the cased vocabulary. The table also shows that Romanian BERT has an order of magnitude fewer unknown tokens than M-BERT on the same text². Table 2 shows a tokenization example for each evaluated model.

Pretraining was performed using the standard BERT recipe. Each model was trained for 1M steps,

²These statistics were computed on a development file extracted from the corpus. This file is composed of 5000 lines that were extracted respecting each individual corpus’s size, reflecting a non-skewed distribution. Neither the vocabulary nor the model were trained on this file.

Model	Frozen		Non-Frozen	
	UPOS	XPOS	UPOS	XPOS
M-BERT (uncased)	95.48	89.84	97.65	95.72
M-BERT (cased)	94.46	89.50	97.87	96.16
Romanian BERT (uncased)	96.55	95.14	98.18	96.84
Romanian BERT (cased)	96.49	95.01	98.00	96.46

Table 4: Simple Universal Dependencies evaluation results.

Model	UPOS	UFeats	Lemmas	LAS
M-BERT (uncased)	97.72	96.54	94.67	87.65
M-BERT (cased)	97.90	96.71	95.2	88.05
Romanian BERT (uncased)	97.91	97.01	94.93	89.61
Romanian BERT (cased)	97.90	97.22	94.88	89.69

Table 5: Joint Universal Dependencies evaluation results.

with the initial 900K trained on a sequence length of 128 and the rest with the maximum length of 512. Figure 1 shows the progress for both models. The sudden increase in loss at the 900K steps mark is due to the switch to the 512 sequence length, but both models quickly recover from it. The models were trained using a batch size of 140 per GPU (for 128 sequence length), and then 20 (for 512 sequence length). The optimizer used was Layer-wise Adaptive Moments optimizer for Batch training (LAMB (You et al., 2019)), with warm-up over the first 1% of steps up to a learning rate of $1e-4$, followed by a decay. Eight Nvidia Volta V100 GPUs with 32GB memory were used, and the pre-training process took around 2 weeks per model.

4 Evaluation

We evaluate Romanian BERT on three downstream tasks:

- **Simple Universal Dependencies:** one model per task, evaluating labeling performance on UPOS (Universal Part-of-Speech) and the XPOS (eXtended Part-of-Speech).
- **Joint Universal Dependencies:** a single model trained jointly on all tasks, evaluating UPOS, UFeats (Universal Features), Lemmas and Dependency Parsing for which we compute the Labeled Attachment Score.
- **Named Entity Recognition:** a single model predicting BIO-style labels.

For each task we compare Romanian BERT with M-BERT, on both cased and uncased versions. To mitigate the effect of the random seed we run each experiment 5 times and report only the mean score.

All results listed here are reproducible by using the evaluation scripts provided on [GitHub](#).

4.1 Simple Universal Dependencies

For this token labeling task we used the Romanian RRT (Barbu Mititelu et al., 2016) dataset from Universal Dependencies (UD), and evaluated the performance of the language models for UPOS and XPOS tagging using the macro-averaged F1, as proposed by Zeman et al. (2018).

Methodology The model itself is straightforward: on top of BERT’s output layer we directly use a linear layer (with a 0.1 fixed dropout) that projects BERT’s outputs into the desired number of classes, depending on the UPOS or XPOS task. Cross-entropy loss is used on the softmaxed linear layer. We perform 2 tests for each UPOS/XPOS task: a frozen test where we train only the added last layer of the model (“freezing” the language model weights), and a full test where all parameters are trainable. The frozen test can give additional insight into a model because its performance now rests more on the pretrained weights rather than the fine-tuned weights, meaning that differences in frozen model performance should be more exaggerated than the fully tuned ones.

Results The results are summarized in Table 4. They show that Romanian BERT outperforms M-BERT on all subtasks, with differences in scores ranging from 0.13% (UPOS non-frozen cased) to 6% (XPOS frozen cased). Moreover, our assumption that the difference between frozen variants will be higher holds.

Surprisingly, the uncased version of

Model	Entity Type	Partial	Strict	Exact
M-BERT (uncased)	84.75	86.06	80.81	83.91
M-BERT (cased)	84.52	86.27	80.6	84.13
Romanian BERT (uncased)	85.53	87.17	82.01	85.26
Romanian BERT (cased)	86.21	87.84	82.54	85.88

Table 6: Named entity recognition evaluation results.

Romanian BERT achieved the highest performance on all experiments, although the cased version should be at something of an advantage on this task thanks to a capital letter indicating proper nouns, etc. We believe these results may be due to better matching between the uncased vocabulary (which necessarily contains more full words) and the RRT corpus.

4.2 Joint Universal Dependencies

For this task we evaluated the language models on the same dataset as in the Simple task, namely RRT. We evaluate the models using to the standard CoNLL shared task evaluation tools (Zeman et al., 2018) and report the scores for UPOS (giving us a comparison to the same UPOS task using a simpler model), UFeats (Universal Features of each word), Lemmas, and the Labeled Attachment Score (LAS).

Methodology Although we evaluated the models on the same dataset, the methodology was different. We use an external tool to perform evaluation: UDify (Kondratyuk, 2019), a Transformer-based model that performs joint training and prediction on every UD subtask in a single step. It implements a prediction layer on top of the contextualized embeddings for each task and a layer-wise dot-product attention that calculates a weighted sum for all intermediate representations of a token.

Results The results shown in table 5 reflect that scores are rather mixed, with the uncased version of Romanian BERT obtaining the highest UPOS score, but with a very small difference from the cased model. For UFeats the Romanian BERT cased is the clear winner with a score of 97.22%. Somewhat surprisingly, M-BERT cased obtained the highest score on lemma generation. The reasoning for this could be that because M-BERT was trained on multiple languages, including many corpora representing Latin languages (of which Romanian is one), it was able to better model lemmas due to the powerful cross-lingual learning capabilities of the

Transformer³. Finally, the LAS score shows a clear improvement from M-BERT with 87.65% uncased and 88.05% cased to Romanian BERT with 89.61% uncased and 89.69% cased.

4.3 Named Entity Recognition

Romanian BERT was evaluated on RONEC (Dumitrescu and Avram, 2019) - a fine-grained NER corpus. The standard BIO tagging was used and models were evaluated according to (Segura Bedmar et al., 2013), reporting the F1 macro-averaged metrics for entity type, partial, strict and exact matches.

Methodology The methodology used for this task is identical to the one used for the Simple Universal Dependencies task. However, we do not evaluate the models on their frozen versions.

Results Table 6 summarizes NER results. Unsurprisingly, we find that Romanian BERT cased has the best performance on this task, improving on M-BERT scores with 0.78% to 1.96% on all metrics. The uncased version of Romanian BERT placed second at the evaluation, outperforming both cased and uncased versions of M-BERT.

5 Conclusions

We have introduced the first BERT model trained solely on a 15GB Romanian corpus, obtained by a thorough clean of OSCAR, OPUS and Wikipedia sub-corpora. We have shown that Romanian BERT outperforms the only available model that works on Romanian, the multilingual M-BERT.

As the current corpus is better cleaned, more text added, tweaks to improve vocabulary coverage are performed, new versions of BERT as well as future models will be released to the open source domain⁴.

³This particular result will lead us to investigate the performance of a multilingual Transformer trained on the Latin languages family

⁴<https://github.com/dumitrescustefan/Romanian-Transformers>

References

- Verginica Barbu Mititelu, Radu Ion, Radu Simionescu, Elena Irimia, and Ceneal-Augusto Perez. 2016. The romanian treebank annotated according to universal dependencies. In *Proceedings of the tenth international conference on natural language processing (hrta2016)*.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. [What does bert look at? an analysis of bert’s attention.](#)
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [Electra: Pre-training text encoders as discriminators rather than generators.](#)
- 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*.
- Daniel Kondratyuk. 2019. 75 languages, 1 model: Parsing universal dependencies universally. *arXiv preprint arXiv:1904.02099*.
- Guillaume Lample and Alexis Conneau. 2019. [Cross-lingual language model pretraining.](#)
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. [Albert: A lite bert for self-supervised learning of language representations.](#)
- 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.](#)
- Paul Michel, Omer Levy, and Graham Neubig. 2019. [Are sixteen heads really better than one?](#)
- 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)*, Cardiff, United Kingdom.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. [A primer in bertology: What we know about how bert works.](#)
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. [Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.](#)
- Isabel Segura Bedmar, Paloma Martínez, and María Herrero Zazo. 2013. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). Association for Computational Linguistics.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. [Bert rediscovers the classical nlp pipeline.](#)
- Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *LREC*.
- 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.
- 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.](#)
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. [Xlnet: Generalized autoregressive pretraining for language understanding.](#)
- Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2019. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*.
- Daniel Zeman, Jan Hajic, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 1–21.