computers

MDPI

Article

A Lite Romanian BERT: ALR-BERT

Dragoş Constantin Nicolae [1,*], Rohan Kumar Yadav [2] and Dan Tufiş [1]

[1] Research Institute for Artificial Intelligence, Romanian Academy, 050711 Bucharest, Romania; tufis@racai.ro
[2] Department of Information and Communication, University of Agder, 4604 Grimstad, Norway; rohan.k.yadav@uia.no
* dragosnicolae555@gmail.com

Abstract: Large-scale pre-trained language representation and its promising performance in various downstream applications have become an area of interest in the field of natural language processing (NLP). There has been huge interest in further increasing the model's size in order to outperform the best previously obtained performances. However, at some point, increasing the model's parameters may lead to reaching its saturation point due to the limited capacity of GPU/TPU. In addition to this, such models are mostly available in English or a shared multilingual structure. Hence, in this paper, we propose a lite BERT trained on a large corpus solely in the Romanian language, which we called "A Lite Romanian BERT (ALR-BERT)". Based on comprehensive empirical results, ALR-BERT produces models that scale far better than the original Romanian BERT. Alongside presenting the performance on downstream tasks, we detail the analysis of the training process and its parameters. We also intend to distribute our code and model as an open source together with the downstream task.

Keywords: BERT; transformers; ALBERT; NLP; Romanian

check for updates

Citation: Nicolae, D.C.; Yadav, R.K.; Tufiş, D. A Lite Romanian BERT: ALR-BERT. *Computers* **2022**, *11*, 57. https://doi.org/10.3390/computers11040057

Academic Editor: Fernando Bobillo

Received: 3 February 2022
Accepted: 11 April 2022
Published: 15 April 2022

1. Introduction

Pre-trained language models are now becoming an important model in the development of current natural language processing (NLP) applications. Recurrent neural nets, such as the LSTM, have previously dominated sequence-to-sequence (seq2seq) [1] modeling for natural languages, achieving record results for core language-understanding tasks. However, a revolution started in NLP with the introduction of the first transformer-based model [2] that demonstrated a significant increase in state-of-the-art results compared to those previously obtained and considered the best. Since its development, the traditional model such as that used in the recurrent neural network was reconsidered and it was opted to fuse the attention process with it in order to relate the different positions in a sequence to then compute a representation of the input.

Bidirectional Encoder Representations from Transformers (BERT) is a recent model developed by Devlin et al. [3] to improve the unidirectional training of a language model by masked language model (MLM) training objectives. The MLM objective enabled the pre-training of deep bidirectional language encoding by conditioning on both left and right contexts at all levels. In addition to this, BERT also uses next-sentence prediction (NSP) for pre-training language models. NSP is supposed to learn high-level linguistic coherence by predicting whether or not two text segments should appear in the same order as they did in the original text. NSP can also help with downstream NLP tasks such as natural language inference, which necessitates thinking about inter-sentence relationships.

Evidence from the performance of the BERT model demonstrates that large models are significant for achieving state-of-the-art results. However, an obstacle to such pre-trained models is the memory limitation of available hardware. With hundreds of millions or even billions of parameters in current state-of-the-art models, it is easy to fall foul of these restrictions when we try to scale our models. To address these limitations, Lan et al. [4] developed a lite BERT architecture (ALBERT) that has significantly fewer parameters

than the traditional BERT architecture. ALBERT uses two parameter-reduction strategies to overcome the most significant challenges in scaling pre-trained models. Factorized embedding parameterization is the first, while cross-layer parameter sharing is the second. Both strategies dramatically reduce the number of parameters for BERT without sacrificing performance, resulting in increased parameter efficiency. Similar to BERT-large, an ALBERT configuration has 18× fewer parameters and can be trained 1.7× faster.

Since the introduction of BERT, it has achieved state-of-the-art performance on accuracy in natural language understanding tasks such as GLUE [5], MultiNLI [6], SQuAD v1.1 [7], SQuAD v2.0 [8] and CoNLL-2003 NER [9]. ALBERT has achieved new state-of-the-art results on the GLUE and SQuAD benchmarks while having fewer parameters than BERT. It is worth noting that a large network is critical for achieving cutting-edge results on downstream tasks. While BERT is an excellent choice for training a large language model on big corpora, it is challenging to experiment with large BERT models due to memory and computational constraints. Hence, in this paper, we develop and train ALR-BERT, a monolingual ALBERT model for Romanian language understanding.

2. Related Works

Transformer [2] is a sequence transduction model that only relies on the attention process, bypassing any recurrent or convolutional neural network components. Multiple identical encoder and decoder blocks are stacked on top of each other in the transformer architecture. While the encoder gathers linguistic information from the input sequence and builds contextual representations, the decoder provides an output sequence that corresponds to the pair of inputs. Transformer can acquire varying attentions inside a single sequence thanks to multi-head self-attention layers in an encoder block, which reduces the inevitable dragging that occurs during recurrent neural network training.

By providing innovative training approaches, BERT sets itself apart from other language models that anticipate the next word based on prior words. Rather than predicting the next token based on previous tokens, it must predict a word that has been replaced by a special token (MASK). This training technique provides BERT with bidirectionality or access to both left and right information around the target word. As a result, BERT is capable of generating a deep bidirectional representation of the input sequence.

On a wide range of NLP tasks, RoBERTa [10], ALBERT [4] and other variations [11,12] use bidirectional context representation and established state-of-the-art results. BERT is trained using masked language modeling (MLM) and next sentence prediction (NSP) losses. The NSP problem is a binary classification job that predicts whether two segments separated by a special token (SEP) in the original text will follow each other. The goal of the task is to understand the relationship between two sentences so that it may be used on a variety of downstream tasks where the input template is made up of two sentences, such as question answering (QA) and sentence entailment [3].

Recent criticism of NSP has claimed that, because of its sloppy inter-sentential coherence, it does not necessarily help improve downstream task performance [4,10,13]. ALBERT, which is based on BERT's architecture, instead uses a sentence order prediction (SOP) task. Negative examples in the SOP task consist of a pair of sentences from the same document, but with the sentence order reversed, the model must predict whether the order is reversed. ALBERT dramatically reduces the amount of parameters—18-fold fewer for BERT-large—while delivering equivalent or better performance on downstream tasks due to reduced SOP loss and other parameter-reduction approaches [4]. ALR-BERT uses the original ALBERT architecture as a starting point. ALR-BERT is being trained from the ground up on large Romanian corpora gathered online.

3. ALR-BERT: Training the Romanian Language Model Using ALBERT

3.1. CORPUS

ALR-BERT was pre-trained using unannotated texts from three publicly accessible corpora: OPUS, OSCAR and Wikipedia, as used by the RomanianBERT [14]. The details of

the corpus are shown in Table 1 which gives the details about the corpus, number of lines, number of words and the size.

- **OPUS**—OPUS is a collection of translated texts from the Web [15]. It is an open source parallel corpus that was compiled without human intervention. It includes a wide range of text types, including medical prescriptions, legal papers, and movie subtitles. The OPUS corpus comprises about 4 GB of Romanian text in total.
- **OSCAR**—OSCAR, or Open Super-large Crawled ALMAnaCH corpora, is a massive multilingual corpus derived from the Common Crawl corpus using language categorization and filtering [16]. There are approximately 11 GB of text in the Romanian portion. It contains scrambled sentences that have been de-duplicated.
- **Wikipedia**—The Romanian Wikipedia is publicly available for download. We used the Wikipedia dump from February 2020 which included approximately 0.4 GB of content after cleanup.

Table 1. Corpus statistics.

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	60.5 M	0.4 GB
Total	90.2 M	2421.3 M	15.2 GB

3.2. ALR-BERT

ALR-BERT uses two parameter reduction techniques as ALBERT to overcome the most significant challenges in scaling pre-trained models. ALR-BERT is a multi-layer bidirectional transformer encoder that shares ALBERT's factorized embedding parameterization and cross-layer sharing. In factorized embedding parameterization, we separate the size of the hidden layers from the size of vocabulary embedding by decomposing the large vocabulary embedding matrix into two small matrices. This separation makes it easier to increase the hidden size without significantly increasing the vocabulary embeddings' parameter size. The factorization of these parameters is accomplished by decomposing the matrix representing the weights of the word embeddings into two distinct matrices. Instead of directly projecting the one-hot encoded vectors onto the hidden space, they are first projected onto a lower-dimensional embedding space, which is then projected onto the hidden space. The WordPiece embedding size E is tied to the hidden layer size H in BERT and subsequent modeling improvements such as XLNet [13] and RoBERTa [10], i.e., $E \equiv H$. As follows, this decision appears to be suboptimal in terms of both modeling and practicality. WordPiece embeddings are used in modeling to learn context-independent representations, whereas hidden-layer embeddings are used in modeling to learn context-dependent representations. The power of BERT-like representations emerges from the use of the context to provide the signal for learning such context-dependent representations, as shown by experiments with context length [10]. As a result, separating the WordPiece embedding size E from the hidden layer size H allows us to make more efficient use of the total model parameters based on modeling needs, which dictate that $H >> E$. In practice, natural language processing usually necessitates a large vocabulary size V. If $E \equiv H$, then increasing H increases the size of the embedding matrix which has the dimensions $V \times E$. This can easily result in a model with billions of parameters, the vast majority of which are only sparsely updated during training. As a result, for ALR-BERT, we factorize the embedding parameters, dividing them into two smaller matrices. Instead of directly projecting the one-hot vectors into the hidden space of size H, we first project them into a lower dimensional embedding space of size E and then to the hidden space.

Another method for increasing parameter efficiency that we propose is cross-layer parameter sharing, which is similar to ALBERT. There are several methods for sharing parameters, such as sharing only feed-forward network (FFN) parameters across layers or only attention

parameters. For cross-layer sharing, this method prevents the parameter from increasing in proportion to the network's depth. Both techniques significantly reduce the number of parameters for BERT without sacrificing performance, resulting in increased parameter efficiency. In general, [MASK] is used to replace a randomly selected subset of tokens in the supplied text and this MLM technique computes a cross-entropy loss on the prediction of the masked tokens. Here, we evenly choose 15% of the input tokens for potential masking with 80% being replaced with (MASK), leaving 10% unaltered and the rest replaced by randomly selected tokens. In addition to this, SOP is acknowledged for focusing on inter-sentence coherence modeling. When the order of two successive segments from the same text is swapped, the SOP loss is used as a positive example and as a negative example.

The ALR-BERT-base inherits ALBERT-base and features 12 parameter-sharing layers, a 128-dimension embedding size, 768 hidden units, 12 heads, and GELU non-linearities [17]. Masked language modeling (MLM) and sentence order prediction (SOP) losses are the two objectives that ALBERT is pre-trained on. For ALR-BERT, we preserve both these objectives. Rather than a formal analysis of training objectives, the main focus of this work is on the empirical side of the design and pre-training of an ALBERT-based foreign-language model. The full architecture of ALR-BERT was borrowed from the ALBERT model [4].

3.3. Pre-Training

Building a vocabulary on the provided corpus was the first step in the pre-training process. We created cased and uncased vocabularies with 50,000 word pieces using byte-pair encoding (BPE) [14]. The number of characters coverage—was set to 2000; to reduce UNKs, we used a larger character set to cover less frequent chars/symbols. A language model's performance is heavily reliant on its vocabulary. In general, the better the tokenization, the better the model is expected to perform. Hence, we used the vocabulary created by Romanian BERT which encodes a word in approximately 1.4 tokens while multilingual-BERT (M-BERT) can reach up to 2 tokens/word for the cased vocabulary [14]. On the same text, Romanian BERT has an order of magnitude of less unknown tokens than M-BERT. For each assessed model, Table 2 presents a tokenization example. The normal BERT recipe was used during pre-training. Each model was trained for 1 million steps, with the first 900 K on a 128-step sequence. The development of both models is depicted in Figure 1. The models were trained using 40 batches per GPU (for 128-sequence length) and then 20 batches per GPU (for 512-sequence length). Layer-wise adaptive moments optimizer for batch (LAMB) training was utilized, with a warm-up over the first 1% of steps up to a learning rate of 1×10^{-4}, then a decay. Eight NVIDIA Tesla V100 SXM3 with 32 GB memory were used, and the pre-training process took approximately 2 weeks per model.

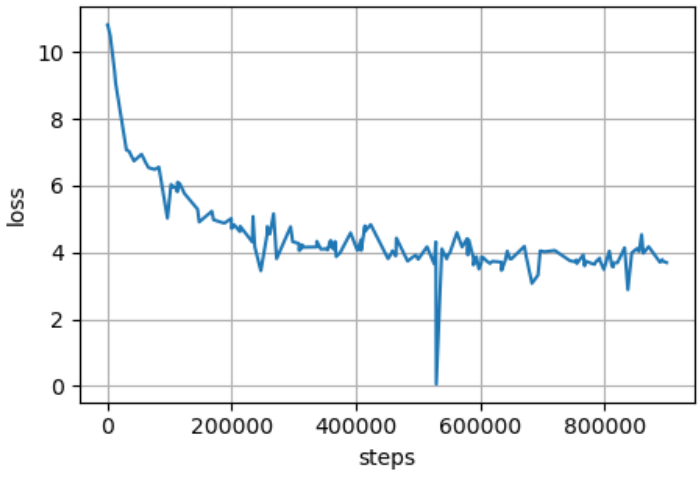

Figure 1. Training loss for the ALR-BERT (cased version).

Table 2. Model tokenization examples [14].

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 ##ti au pl ##eca ##t din C ##rai ##ova spre ##op ##r ##li ##a .
Romanian BERT (uncased)	cinci bicicliti au plecat din craiova spre opr ##lia .
Romanian BERT (cased)	Cinci bicicliti au plecat din Craiova spre o ##p ##r ##lia .

4. Evaluations

Here, we evaluated ALR-BERT on the Simple Universal Dependencies task. There is one model for each task to evaluate the labeling performance on the Universal Part-of-Speech (UPOS) and the Extended Part-of-Speech (XPOS). We compare our proposed ALR-BERT with the Romanian BERT and multilingual BERT using the cased version. To counteract the random seed effect, we repeated each experiment five times and simply provided the mean score. All the codes as well as the link to datasets are provided online at Github.

4.1. Simple Universal Dependencies

For this task, we used the Romanian RRT [18] dataset from Universal Dependencies (UD) and evaluated the performance of the proposed language models for UPOS and XPOS tagging with evaluation metrics being macro-averaged F1 [19].

We used the same evaluation method as that used for Romanian BERT [14] where a linear layer was used on top of our ALR-BERT's output layer with a fixed dropout of 0.1. We also used cross-entropy loss on the softmax linear layer. We then performed multiple tests for each UPOS and XPOS task. On the frozen test, only the last layer of the model was trained while freezing the language model's weights and the second test is performed once all the parameters are trained. In addition to this, we also evaluated the proposed model for morphology-aware labeled attachment score (MLAS). It is a CLAS extension that includes the UPOS tag and a morphological feature evaluation [19]. Finally, we also selected AllTags as an evaluation metric.

4.2. Results

The results obtained from the experiments are detailed in this section. Table 3 shows the performance of UPOS, XPOS, MLAS and AllTags for the frozen test. Here, we can see that the performance of the Romanian BERT excels across all of the datasets, outperforming its main competitor—multilingual BERT. However, given the parameter size and the fact that ALR-BERT is very light compared to the superior models, its accuracy does not drastically degrade. ALR-BERT reached 87.5% and 84.05% in UPOS and XPOS, respectively, which is approximately an 8% reduction in accuracy, which is not very significant given the reduction in parameters in the ALBERT family. However, the performance on MLAS and AllTags is significantly reduced for the frozen test. This may be due to the frozen state of the parameters which restricts the model, preventing its learning from the deep non-linear structure of the dataset.

In the case of the non-frozen test, Table 4 shows that the accuracy of ALR-BERT is very promising and has a very small degradation of accuracy across all four of the selected datasets. The ALR-BERT achieves 95.03% compared to Romanian BERT's 98.26% and M-BERT's 97.95%. Similarly for XPOS, ALR-BERT obtains 89.92% compared to Romanian BERT's 96.96% and M-BERT's 96.12%. Even for MLAS and AllTags, the performance of ALR-BERT was only 6–8 percentage points less than the compared models. This is due to the fact that the parameters are not frozen and are fine-tuned. Given that this performance comes with 70% fewer parameters [4], compared to BERT models, it has higher data throughput. As a result, they can train 1.7 times faster than BERT.

Ablation Studies

The ablation studies show the performance of an AI system by removing certain components to understand the contribution of the component to the overall system. General ablation results were obtained from the ALBERT model [4].

- The ALBERT model states that factorized embeddings perform well where cross-layer parameters were not shared as well as the case where they were shared. Larger embedding sizes provide greater performance in the absence of sharing. At an embedding size of 128 dimensions, speed improvements are satisfied with sharing.
- For cross-layer parameter sharing, the ALBERT model shows various cross-layer parameter sharing analysis: (a) no cross-layer sharing was seen; (b) cross-layer sharing was observed for the feedforward segments only; (c) performing sharing for the attention segments; and (d) performing sharing for all subsegments. It turns out that sharing the parameters for the attention segments is the most effective [4], but sharing the parameters for the feed-forward segments has very little effect. This exemplifies the importance of the attention process in transformer models. However, because all-segment sharing greatly reduces the number of parameters while providing only slightly inferior performance than attention-only sharing, the authors chose to adopt all-segment sharing instead.
- If model uses the NSP technique on an SOP task, then the performance is low. Of course, NSP on NSP performs well, as does SOP on SOP. However, when SOP is run on NSP, it works really well. This implies that SOP catches sentence coherence that NSP may not, and therefore SOP produces a better outcome than NSP.

Table 3. Simple universal dependencies evaluation results on the frozen test.

Model	UPOS	XPOS	MLAS	AllTags
M-BERT (cased)	93.87	89.89	90.01	87.04
Romanian BERT (cased)	95.56	95.35	92.78	93.22
ALR-BERT (cased)	87.38	84.05	79.82	78.82

Table 4. Simple universal dependencies evaluation results on non-frozen test.

Model	UPOS	XPOS	MLAS	AllTags
M-BERT (cased)	97.95	96.12	96.61	95.69
Romanian BERT (cased)	98.24	96.96	97.08	96.60
ALR-BERT (cased)	95.03	89.92	91.96	88.75

5. Conclusions

The present research introduced a lite Romanian BERT (ALR-BERT) to a huge corpus consisting of OSCAR, OPUS and Wikipedia sub-corpora. We showed that ALR-BERT does not drastically reduce accuracy, which is a promising development towards reducing model size. This model aims to fill the gap in the research and the practical implementation of NLP applications by reducing the model size without a big drop in accuracy.

In future work, we would like to explore some additional preprocessing that would boost the accuracy to the level of the Romanian BERT and maintaining its minimal model size.

Author Contributions: Conceptualization, D.C.N. and R.K.Y.; methodology, D.C.N.; software, D.C.N.; validation, D.C.N. and R.K.Y.; formal analysis, D.C.N. and D.T.; investigation, D.C.N.; resources, R.K.Y.; writing—original draft preparation, D.C.N. and R.K.Y.; writing—review and editing, D.C.N. and D.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In Proceedings of the NIPS 2014, Montreal, QC, Canada, 8–13 December 2014.
2. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. *arXiv* **2017**, arXiv:1706.03762.
3. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL 2019, Minneapolis, MN, USA, 2–7 June 2019.
4. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv* **2020**, arXiv:1909.11942.
5. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*; Association for Computational Linguistics: Brussels, Belgium, November 2018.
6. Williams, A.; Nangia, N.; Bowman, S.R. *A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference*; NAACL: New Orleans, LA, USA, 2018.
7. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; Association for Computational Linguistics: Austin, TX, USA, 2016; pp. 2383–2392.
8. Rajpurkar, P.; Jia, R.; Liang, P. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*; Association for Computational Linguistics: Melbourne, Australia, July 2018; pp. 784–789.
9. Tjong Kim Sang, E.F.; De Meulder, F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In Proceedings of the Seventh Conference on Natural Language Learning at HLT—-NAACL Edmonton, AL, Canada, 31 May–1 June 2003; pp. 142–147.
10. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv: 1907.11692.
11. Joshi, M.; Chen, D.; Liu, Y.; Weld, D.S.; Zettlemoyer, L.; Levy, O. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 64–77. [CrossRef]
12. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z.; Wang, S.; Hu, G. Pre-Training with Whole Word Masking for Chinese BERT. *arXiv* **2019**, arXiv:1906.08101.
13. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.G.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In Proceedings of the NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019.
14. Dumitrescu, S.; Avram, A.M.; Pyysalo, S. The birth of Romanian BERT. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 5–10 July 2020; pp. 4324–4328.
15. Tiedemann, J. Parallel Data, Tools and Interfaces in OPUS. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, 21–27 May 2012; European Language Resources Association (ELRA): Luxembourg, 2012; pp. 2214–2218.
16. Suarez, P.J.O.; Sagot, B.; Romary, L. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. In Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019, Cardiff, UK, 22 July 2019; Leibniz-Institut für Deutsche Sprache: Mannheim, Germany, 2019; pp. 9–16.
17. Hendrycks, D.; Gimpel, K. Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. *arXiv* **2016**, arXiv:1606.08415.
18. Mititelu, V.B.; Ionn, R.; Simionescu, R.; Irimia, E.; Perez, C.A. The romanian treebank annotated according to universal dependencies. In Proceedings of the Tenth International Conference on Natural Language Processing (HRTAL'16), Dubrovnik, Croatia, 29 September–1 October 2016.
19. Zeman, D.; Hajic, J.; Popel, M.; Potthast, M.; Straka, M.; Ginter, F.; Nivre, J.; Petrov, S. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies; In Proceedings of the CoNLL 2018, Brussels, Belgium, 31 October–1 November 2019.