Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2020”

Moscow, June 17–20, 2020

TAGGING AND PARSING OF MULTIDOMAIN COLLECTIONS

Sorokin A. A. (alexey.sorokin@list.ru)

Moscow Institute of Physics and Technology, Neural Networks and Deep Learning Lab, Dolgoprudny, Russia;
Moscow State University, Faculty of Mathematics and Mechanics, Moscow, Russia

Smurov I. M. (ivan.smurov@abbyy.com)

ABBYY, Moscow, Russia;
Moscow Institute of Physics and Technology, Computer Vision and Natural Language Processing Laboratory (ABBYY Lab), Dolgoprudny, Russia

Kirianov D. P. (denkirjanov@gmail.com)

SberDevices, Moscow, Russia

In this paper we describe our submission to GramEval2020 competition on morphological tagging, lemmatization and dependency parsing.

Our model uses biaffine attention over the BERT representations. The main feature of our work is the extensive usage of language model, tagger and parser fine-tuning on several distinct genres and the implementation of genre classifier. To deal with dataset idiosyncrasies we also extensively apply handwritten rules.

Our model took second place in the overall model performance scoring 90.8 aggregate measure over all 4 tasks.

Keywords: morphological tagging, lemmatization, dependency parsing, domain adaptation, fine-tuning, BERT

DOI: 10.28995/2075-7182-2020-19-670-683

МОРФОЛОГИЧЕСКИЙ И СИНТАКСИЧЕСКИЙ АНАЛИЗ МУЛЬТИЖАНРОВЫХ ТЕКСТОВЫХ КОЛЛЕКЦИЙ

Сорокин А. А. (alexey.sorokin@list.ru)

Московский Физико-технический Институт, Лаборатория нейронных систем и глубокого обучения, Долгопрудный, Россия;
Московский Государственный Университет, механико-математический факультет, Москва, Россия

Смуров И. М. (ivan.smurov@abbyy.com)

ABBYY, Москва, Россия;
Московский Физико-технический Институт, Лаборатория компьютерного зрения и обработки естественного языка (ABBYY Lab), Долгопрудный, Россия

Кирьянов Д. П. (denkirjanov@gmail.com)

SberDevices, Москва, Россия

В данной статье описана наша модель, использованная в соревновании GramEval2020 по морфологическому анализу, лемматизации и синтаксическому анализу. Наша модель основана на бифинном механизме внимания и архитектуре БЕРТ. Её отличительной чертой является использование отдельной модели БЕРТ для каждого жанра и настройка базовой модели на доменных данных, а также использование правил для унификации разметки.

Наша модель заняла второе место в соревновании, показав среднее качество 90,8% по 4 заданиям и 6 предметным областям, в то время как результат победителя составил 91,7%.

Ключевые слова: морфологический анализ, лемматизация, синтаксический анализ, дообучение, БЕРТ

1. Introduction

Automatic processing of morphology and syntax have been a part of Natural Language Processing (NLP) for several decades. Introduction of end-to-end deep learning pipelines [7], embeddings pretraining ([17], [18]), and char-level features [16] have all contributed to rapid improvement of NLP in general and grammatical features extraction systems in particular. This trend was enhanced by a recent introduction of pretrained language models and context-dependent embeddings such as ELMo [22] and BERT [8] leading to a drastic improvement for an overwhelming majority of NLP tasks including grammatical tagging and parsing.

Dependency parsing has seen a similar evolution of its own in the last half-decade. This process was kick-started by [5] who introduced deep learning into transition-based parsing [20]. Two years later [12] were able to successfully utilize RNNs in both transition-based and graph-based dependency parsing. While their work allowed for a remarkable increase of graph-based parsing quality it wasn't until [9] model before graph-based parsing dethroned transition-based parsing as a state-of-the-art (SOTA) parsing approach. Most models introduced after 2017 follow the path tread by Dozat and Manning and implement biaffine attention with different feature sets, first utilizing ELMo and most recently BERT.

Currently the best performing dependency parsers for English are immediate successors of HPSG model introduced in [24]. The most important feature of this family of parsers is joint learning on dependency and constituency trees. Unfortunately, this makes utilizing HPSG-style parsing for Russian an extremely complicated task since there are no publicly avaliable annotated corpora or constituency parsers. While we would like to explore the possibility of utilizing proprietary parsers such as Compreno [1], for now this remains for future work. Given these considerations we decided to base our model on biaffine attention of Dozat and Manning [9] with BERT-based token features. This approach is common in modern NLP and is utilized, e.g., in [13]

Section 2 describes GramEval-2020 shared task and the corpora made available for it. **Section 3** gives an overview of our model. **Section 4** describes the process of training and provides the evaluation results. **Section 5** contains analysis of our model performance as well as the discussion on the representativeness of the Shared Task results for the processing of Russian morphology and syntax. Finally, **Section 6** provides conclusion and outlines our plans for future work.

2. GramEval-2020 and Corpus Analysis

GramEval-2020 [21] is a shared task on part-of-speech (POS) and full morphological tagging, lemmatization and dependency parsing of Russian texts. Parsing was scored with labeled attachment score (LAS) while the other three tasks with accuracy. The participants systems were ranked by an aggregate measure on all four tasks.

For training GramEval-2020 organizers provided not a single corpus, but rather a collection of several disjoint corpora of different origins and genres.

Train set consisted of the following subcorpora:

1. SynTagRus [10] corpus of dependency parses from UD [19] (≈ 62k sentences). Mostly contains texts of general domain.
2. MorphoRuEval2017 [23] morphologically labeled corpus with semi-automatic syntactic annotation. Contains texts of general domain and of social networks from GIKRYA.
3. Poetry corpus (≈ 0.9k sentences). Contains poetic texts from Taiga.
4. Social networks corpus (≈ 2.3k sentences). Contains social media texts from Taiga.
5. Wikipedia texts from GSD (about 5k sentences). Contains texts of general domain and technical texts.

6. XVII century corpus(≈ 1.2k sentences). Contains Middle Russian texts both in original and in an adapted orthography (where symbols not present in modern Russian were substituted by their closest analogues e. g. ” was substituted with ‘E’; the rest of spelling remained unchanged). In total, approximately 60% texts of train set were in original orthography and 40% in adapted.

Development set consisted of subcorpora 3–6 as well as news subcorpora from Lenta.ru (each subcorpus contained 40–70 sentences). Test set included the same sources as the development one and a fiction subcorpus.

3. Model Overview

3.1. Pipeline

The Shared Task data clearly consists of 4 isolated segments whose syntax may differ significantly: social media, poetry, historical (XVII century) texts and general domain. For the XVII century subcorpus its morphology and even graphics is also specific. We expect that there is no single tagger and parser that works for all domains equally well. Therefore we apply a separate model to each domain. Consequently, we use the following pipeline (see subsequent subsections for the description of its components):

1. The classifier predicts a domain label given the input sentence.
2. The morphological tagger outputs a sequence of morphological tags given the tokenized sentence.
3. The lemmatizer yields the source form of the word based on its tag and the word itself.
4. The dependency parser reconstructs the dependency tree given the tokenized sentence. The parser does not take morphological tags into account.
5. Several rule-based postprocessors modify tag and lemma to match the annotation standards. Some of the postprocessors use only the word, its lemma and tag, several other are domain-specific and hence use the class label as well. Some postprocessors also utilize tags of the words in the neighbourhood to fill the missing morphological features.

3.2. Classifier

We experimented with three different architectures of domain classifier: fastText [3] classifier, BERT [8] classifier and a classifier based on logistic regression over character ngrams. The two latter models have shown comparable results (approximately 91% macro F-score). Since logistic regression is both less computationally expensive and easier to interpret and tune we decided to choose the latter one.

We trained the domain classifier on the provided training data. Since the development set is too small to measure performance on it, we left one-quarter of the original training set as the held-out data (this approach was used only to detect the genres).

3.3. Morphological tagger

Both part-of-speech and full morphological tagging can be considered as sequence labeling tasks. Thus one can approach both tasks using standard sequence labeling techniques. In our model we used BERTs for tagging in the same way as [8] treats named-entity recognition task (see figure below).

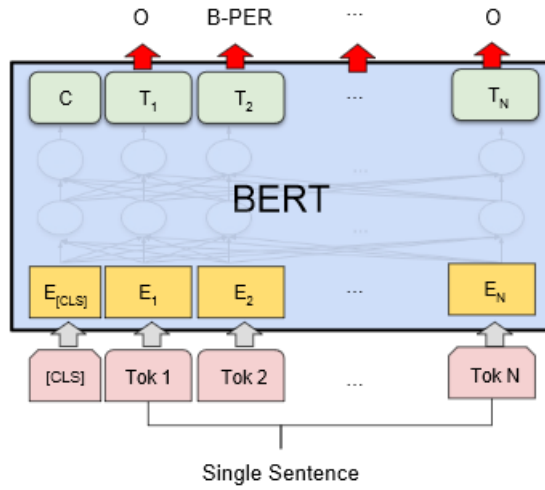


Figure 1: Illustration from [8]

3.4. Lemmatizer

In order to generate possible lemmas we used PyMorphy 2.0 [14] for initial lemma generation and several postprocessing rules to deal with format mismatch. Since PyMorphy outputs several possible variants, we select the one corresponding to the tag predicted by the morphological tagger using the DeepPavlov library [4].

3.5. Dependency parser

As mentioned earlier, for dependency parsing we used biaffine attention network [9] built over BERT contextualized word features (fine-tuned separately for each genre).

Going into more detail, given a contextualized token representation w_i, we generate 4 vectors $head_i^{(arc)}$, $dep_i^{(arc)}$, $head_i^{(rel)}$ and $dep_i^{(rel)}$ using multi-layered perceptrons. $head_i^{(arc)}$ and $dep_i^{(rel)}$ are used to predict the probability of a token being an arc head and arc dependent respectively. In order to generate score for arc from token w_i to token w_j we use biaffine attention as follows:

$$s_{ij}^{(arc)} = head_i^{(arc)} \cdot A \cdot dep_j^{(arc)} + head_i^{(arc)} \cdot b.$$

These scores were transformed to probabilities using standard SoftMax layer. These probabilities are used to produce a maximum spanning tree i. e. dependency tree with the highest probability using Chu-Liu/Edmonds algorithm [6], [11].

Given the dependencies, $head_i^{(rel)}$ and $dep_i^{(rel)}$ can be interpreted and combined in a similar way to generate a probability distribution over all possible dependency labels using a biaffine classifier.

3.6. Postprocessors

We apply several rule-based postprocessors. There main goal is to manually transform the outputs of all models to the same format. For example, not all datasets annotate Animacy of the adjectives. This information can be copied from the parent node of the adjective, however, this requires the precence of syntactic tree. Other postprocessors use only the word itself or/and its lemma/tag. Our final model contains the following postprocessing stages:

1. Emoji postprocessor that uses the Emoji library[1].
2. Adjective animacy postprocessor.
3. Pronoun *что/который* "what/which" postprocessor, that fills the gender/number information for these words. It finds the antecedent of the pronoun by traversing the dependency tree using rule-based instructions and copies the relevant feature values from it.
4. Digit postprocessor. It explores whether the digit satisfies some frequent patterns (such as *3 июня* "3rd of June" or 1917–1920) and calculates the required features (NumType and Case/Number/Gender when applicable).

4. Models and results

4.1. Training approaches

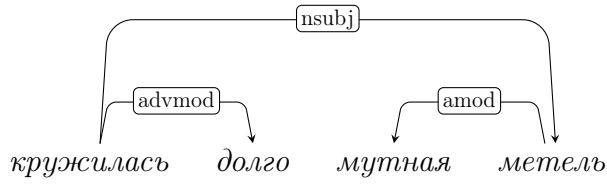
In our study we apply several approaches to model training including

1. Standard supervised model training. In this case we initialize the embedder with the weights of pretrained BERT language model and train the whole network together. This mostly affect the task-specific head layers, however, the weights of the embedder are also altered.
2. BERT language model unsupervised finetuning [2], [15]. If we suspect that dataset domain (e. g., poetry) does not match the domain BERT was trained on, we additionally train the masked language model on the domain-specific raw data.

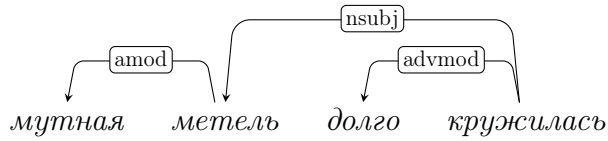
[1] https://pypi.org/project/emoji/

3. Supervised model finetuning. We initialize the model weights using the weights of basic supervised model (see Approach 1) and tune them on the domain-specific annotated data. We follow this strategy when there is not enough such data for learning the weights from scratch, however, the basic model seems to be less suitable due to data peculiarity.

4. Word order adaptation: the word order in verses often differs from the one in formal speech. Consider the verse



its "formal" version would be



More precisely, though the word order in Russian is flexible, for most syntactic dependencies the relative positions of the head and the dependent are rather predictable. In particular, the subject (nsubj) usually precedes the verb, the adjective modifier precedes the corresponding noun, etc. However, for poetic texts these patterns are less strict.

Hence, to make the formal prose more similar to the poetry, we randomly switch the ordering of the head and the dependent for those syntactic relations whose dependent-head order in poetic texts deviates from the one in formal corpus. The switch probability is set to match the order distribution in the poetic training set. Note that this dataset is not large enough to train the model only on it without utilizing the general domain in some fashion.

Graphic adaptation. The most challenging domain in the test data is the XVII century Russian language. Since its alphabet contains several symbols that are not present in the modern Russian, BERT tokenizer is not able to divide such words to subtokens in a proper way. Consider the word *человкъ* (the man), its tokenization will be

челов/##/##къ.

On the contrary, the modern Russian spelling of the same word (*человек*) produces a single subtoken for the whole word. It implies that after tokenization a word in old orthography loses its similarity to the corresponding word in modern orthography, hence the model is unable to utilize the unsupervised knowledge stored in subtoken embeddings. To overcome this obstacle we apply several handwritten rules, such as

- Removal of word-final -*ъ*.
- → *е*, → *и* and analogous modifications.
- Changing word-final -*ти* to -*ть* in verb infinitives, etc.

4.2. Model selection and evaluation

In this section we describe how we select the optimal model for different parts of the dataset. The selection procedure for the tagger and lemmatizer differs from the one for the parser, so we discuss them separately. However, they share several common steps, which we list below.

1. First, we found that data annotation is inconsistent through the training data, as different segments of it were annotated using different annotation standards. Moreover, inspecting the training data we found many errors and artefacts of automatic annotation, therefore we decided to rely only on some subsets of the dataset. Namely, we selected the SynTagRus v2.5 dataset to be the only training source for our basic model[2]. The remaining parts of the training set are used only for validation and finetuning purposes.

2. Since our classifier has 4 possible domain labels ('17cent', 'poetry', 'social' and 'other'), we picked a subcorpus for each of these categories. For first three categories the choice is unambigious, while the performance on 'other' was evaluated on Wikipedia GSD subcorpus and Lenta News subcorpus from 2017 MorphoRuEval competition [23].

3. Since we have no access to the correct annotation of the test set, we make conclusions using the official training data, as only the SynTagRus part of the training set was used for pure training. We used several parts of the training set for model finetuning, in this case we report performance on the development set.

4. During our preliminary experiments we found that altering BERT embedders and tuning the model has no significant effect on the tagger performance, therefore we performed experiments on BERT and model finetuning only for the syntactic parser.

5. All our models are based on ruBERT model [15] from DeepPavlov library [4]. This model was obtained by funetuning the multilingual BERT [8] on Russian language data.

4.3. Tagger performance

We present the performance of our basic model on 5 mentioned domains: '17cent', 'poetry', 'social', 'wiki' and 'news' (the two latter being the part of 'other'). We also present the scores of the finetuned 17th century model on '17 cent' segment. For comparison purposes we also train another variant of the basic model on the unified training set and evaluate it on development set. We present scores on development set only.

[2] The organizers informed us that SynTagRus guidelines does not match the annotation of the test set. However, they did not answer if there was any other subset better corresponding to the annotation of the test data. Using SynTagRus, we at least expect different parts of our training set not to contradict each other.

Table 1: The results of different morphological taggers on the training and development set. We report POS accuracy(P), lemmatization accuracy (L) and morphological feature recall (F) using the official evaluation script.

	wiki			news			poetry			social			17cent		
Model	F	L	P	F	L	P	F	L	P	F	L	P	F	L	P
Basic	96.1	96.6	94.5	95.9	98.4	96.7	94.1	97.1	92.8	95.1	98.4	93.8	87.1	51.7	92.8
1 + 17cent rules	—	—	—	—	—	—	—	—	—	—	—	—	92.6	86.2	94.4
2 + 17cent finetuned	—	—	—	—	—	—	—	—	—	—	—	—	96.0	86.8	98.0
joint	98.4	96.9	98.2	97.8	98.8	98.4	95.6	97.1	95.0	97.2	98.3	95.7	94.2	51.2	96.6
+ 17cent rules	—	—	—	—	—	—	—	—	—	—	—	—	94.2	85.4	96.7

We observe that data-specific rules for 17 century data actually help a lot, finetuning on more 17 century data also improves the model drastically. More surprising is the fact that the joint model beats the one trained only on SynTagRus by a notable margin. However, this question requires further investigation.

4.4. Parser performance

We expect the parser to depend more severely from the training domain. Therefore here we perform a much more detailed comparison, which is performed in several stages. First, we want to select an optimal BERT embedder for each of the domains. We test three BERT models.

1. The default ruBERT model from DeepPavlov library[3].
2. The Conversational BERT model from DeepPavlov library[4] finetuned on social network data from Taiga corpus.
3. The StihBERT model, finetuned on poetry data from Taiga corpus.

All these models are trained only on SynTagRus data. The results are presented in Table 2.

Table 2: The effect of different BERT embedders on syntactic parsing. We report Labeled Attachment Score (LAS) for training and development set.

	wiki		news		poetry		social		17cent	
Model	T	D	T	D	T	D	T	D	T	D
ruBERT	83.8	86.7	92.9	92.9	69.1	79.0	77.1	82.8	59.0	72.8
ConvBERT	83.3	86.5	93.0	92.0	71.4	80.2	78.4	83.2	54.1	72.8
StihBERT	—	—	—	—	72.5	81.5	—	—	—	—

[3] http://docs.deeppavlov.ai/en/master/features/models/bert.html, http://files.deeppavlov.ai/deeppavlov_data/bert/rubert_cased_L-12_H-768_A-12_v2.tar.gz

[4] http://files.deeppavlov.ai/deeppavlov_data/bert/ru_conversational_cased_L-12_H-768_A-12.tar.gz

We observe that using domain-specific BERT improves results on social and poetry subsets, as expected. Consequently, we decide to use StihBERT for poetic data, ConvBERT for the social media and the basic ruBERT for the remaining.

We also evaluate the effect of fine-tuning on heritage, social and poetic data (for 'other' domain the results are controversial). Table 3 contains the results for 'social' and 'poetry' domains. "No finetuning" means choosing the best BERT for a given domain between the models evaluated in Table 2. We also present the results on the 'poetry' data for the model trained on the dataset with switched order of heads and dependencies, as discussed in Subsection 4.1. For comparison we give the scores of the 'joint' model, as it can be viewed as the model fine-tuned on the concatenation of all training data available.

Table 3: The effect of fine-tuning on syntactic parsing. We report Labeled Attachment Score (LAS) for different parts of the development set.

	wiki		news		poetry		social	
Model	T	D	T	D	T	D	T	D
No finetuning	83.8	86.7	92.9	92.9	72.5	81.5	78.4	83.2
+FT(social)	—	87.0	—	91.7	—	78.9	—	85.7
+FT (poetry)	—	—	—	—	—	81.9	—	—
+switch	—	—	—	—	—	82.4	—	—
joint	—	87.0	—	91.2	—	74.0	—	81.5

We observe the positive effect of fine-tuning. Also note that fine-tuning the model on 'social' data decreases its performance on other domains. In contrast to morphological tagging, the model trained on joint data has significantly lower performance. We suppose that one of the reasons may be inconsistent annotation of syntactic phenomena in different training subsets.

We also compare the basic model with the fine-tuned one of the 17 century data. The results are given in Table 4. Here we again observe the positive influence of model fine-tuning.

Table 4: The effect of fine-tuning on syntactic parsing on 17 century data

	17 cent	
Model	T	D
No finetuning	59.0	72.8
+rules	63.5	73.6
+rules+FT	—	85.8
joint	—	78.3
+rules	—	78.3

5. Analysis and Discussion

Table 5 and Table 6 contain the official GramEval-2020 results.

Table 5: GramEval results on historic texts (17) and fiction (fict)

corpus	all	17				fict			
team	overall	POS	morph	lemmas	LAS	POS	morph	lemmas	LAS
qbic	**91.6**	**96.3**	**93.0**	78.3	**66.5**	**98.0**	**98.8**	**98.1**	**89.6**
ADVance (our model)	90.8	96.0	**93.0**	**79.7**	61.9	**98.0**	98.6	97.7	87.0
lima	87.9	93.5	89.6	61.1	55.5	97.6	97.9	93.7	85.1
vocative	85.2	87.1	79.4	58.3	50.0	97.5	94.8	96.2	82.7

Table 6: GramEval results on news and poetry (poet)

corpus	all	news				poet			
team	overall	POS	morph	lemmas	LAS	POS	morph	lemmas	LAS
qbic	**91.6**	96.7	98.1	**98.3**	**91.3**	**96.9**	**96.7**	**95.4**	**81.4**
ADVance (our model)	90.8	96.5	**98.2**	98.2	91.2	96.1	96.0	95.3	78.1
lima	87.9	**97.2**	96.7	95.0	84.4	95.8	95.6	91.3	72.6
vocative	85.2	96.6	94.5	95.5	83.5	92.3	89.8	93.9	66.0

Table 7: GramEval results on social media (soc) and wikipedia (wiki)

corpus	all	soc				wiki			
team	overall	POS	morph	lemmas	LAS	POS	morph	lemmas	LAS
qbic	**91.6**	**94.8**	94.7	**96.0**	**80.7**	**92.7**	94.4	**93.6**	**78.1**
ADVance (our model)	90.8	93.8	**95.9**	95.4	78.5	92.2	92.3	92.2	76.1
lima	87.9	93.7	95.3	95.3	71.3	92.5	**96.8**	92.3	69.8
vocative	85.2	91.8	90.0	95.5	66.0	91.0	90.5	91.6	69.5

Analyzing them, one can notice that our model and the winner's model significantly outperform the other two models. The main difference of the top two models is the usage of BERT. As expected using BERT provides for a significant advantage.

The overall gap between our model and the winner's model is relatively small but consistent across domains. This is mostly due to dependency parsing performance: while on other tasks the scores of these two systems are comparable, our LAS scores are considerably lower on all corpora but news.

After the release of all systems into the open source we have spent some time analyzing the winner's model. Both models use similar architecture and the same ruBERT embedder, however, there is a number of differences in training procedure. So far we were not able to isolate the decisive one, but we would like to explore it more in future work.

5.1. Data quality

Before criticising the datasets provided for the Shared Task, we would like to deeply thank the organizers for their work, which is in important contribution for future studies on computational syntax and morphology of Russian. However, the quality of the provided training data makes the conclusions not so reliable as they could be.

First, the annotation of some morphological phenomena is inconsistent. For example, the label assigned to foreign proper nouns is NOUN, PROPN or SYM depending from the segment. This problem holds for the annotation of proper nouns in general, a word may have controversial labels even in consecutive sentences.

Second, the annotation is inconsistent even inside segments. Namely, while the training data for XVII century contains a significant fraction of texts in original orthography, the development set is entirely in modern (adapted) one. Many specific syntactic relations (e.g., nsubj:pass or det) are occasionally replaced with their more general analogues (nsubj and amod). The same problem holds for fixed constructions, e.g. complex prepositions as *со стороны* "by", which are not annotated in most of the training corpora, being present only in one of them. Additionally, the annotation of punctuation is also inconsistent, which produces many spirious errors that do not reflect the actual performance of the model.

Last but not the least, the significant amount of data is annotated automatically using a model of rather low quality. It sometimes yield nonse errors, for example, the noun *джакузи* has the lemma **джакузить* in the training set.

6. Conclusion and Future Work

We have presented ADVance—a system performing part-of-speech and full morphological tagging, lemmatization and dependency parsing for Russian. Our system has participated in GramEval-2020 shared task and was able to reach second place. Our morphological tagger uses BERT as contextualized embedder and the parser system is based on biaffine attention over BERT representations. We release our system in open source[5].

Our main scientific contribution is the relative success of domain adaptation and fine-tuning approaches, that goes in line with previous studies. However, the results on dependency parsing on challenging poetry and XVII century domains are well below the scores for more formal texts. Additionally, this scores are lower than the ones reported for Universal Dependencies datasets, where the Basic version of our model achieves LAS over 93%. That is partially due to train-test annotation mismatch, however, this is a common real world situation. We hope that our study will help to shed light on practical aspects of training morphological and syntactic analyzers on real-world data with imperfect annotation.

[5] https://github.com/AlexeySorokin/GramEval2020

7. Acknowledgements

We thank the organizers of Shared Task Olga Lyashevskaya and Tatiana Shavrina for providing the data and holding the competition. We are also grateful to members of DeepPavlov team for their assistance in BERT fine-tuning. We thank the anonymous reviewers whose valuable comments helped to improve the paper.

References

1. *Anisimovich, K. et al.:* Syntactic and semantic parser based on abbyy compreno linguistic technologies. In: Computational linguistics and intellectual technologies: Proceedings of the international conference "dialog" [komp'iuternaia lingvistika i intellektual'nye tehnologii: Trudy mezhdunarodnoj konferentsii "dialog"]. pp. 90–103, Bekasovo, Russia (2012).
2. *Arkhipov, M. et al.:* Tuning multilingual transformers for language-specific named entity recognition. In: Proceedings of the 7th workshop on balto-slavic natural language processing. pp. 89–93 (2019).
3. *Bojanowski, P. et al.:* Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics. 5, 135–146 (2017).
4. *Burtsev, M. et al.:* Deeppavlov: Open-source library for dialogue systems. In: Proceedings of acl 2018, system demonstrations. pp. 122–127 (2018).
5. *Chen, D., Manning, C. D.:* A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp). pp. 740–750 (2014).
6. *Chu, Y.-J., Liu, T.-H.:* On the shortest arborescence of a directed graph. Scientia Sinica. 14, 1396–1400 (1965).
7. *Collobert, R. et al.:* Natural language processing (almost) from scratch. Journal of machine learning research. 12, Aug, 2493–2537 (2011).
8. *Devlin, J. et al.:* BERT: pre-training of deep bidirectional transformers for language understanding. CoRR. abs/1810.04805, (2018).
9. *Dozat, T., Manning, C. D.:* Deep biaffine attention for neural dependency parsing. arXiv preprint arXiv:1611.01734. (2016).
10. *Dyachenko, P. et al.:* Sovremennoe sostoyanie gluboko annotirovannogo korpusa tekstov russkogo yazyka (syntagrus). [The current state of the deeply annotated corpus of russian texts (syntagrus)]. Trudy Instituta Russkogo Yazyka im. V. V. Vinogradova. 6, 272–300 (2015).
11. *Edmonds, J.:* Optimum branchings. Journal of Research of the national Bureau of Standards B. 71, 4, 233–240 (1967).
12. *Kiperwasser, E., Goldberg, Y.:* Simple and accurate dependency parsing using bidirectional LSTM feature representations. Transactions of the Association for Computational Linguistics. 4, 313–327 (2016).
13. *Kondratyuk, D.:* 75 languages, 1 model: Parsing universal dependencies universally. arXiv preprint arXiv:1904.02099. (2019).
14. *Korobov, M.:* Morphological analyzer and generator for russian and ukrainian languages. 542, 320–332 (2015).

15. *Kuratov, Y., Arkhipov, M.:* Adaptation of deep bidirectional multilingual transformers for russian language. arXiv preprint arXiv:1905.07213. (2019).
16. *Lample, G. et al.:* Neural architectures for named entity recognition. In: Proceedings of the 2016 conference of the north American chapter of the association for computational linguistics: Human language technologies. pp. 260–270 Association for Computational Linguistics, San Diego, California (2016).
17. *Mikolov, T. et al.:* Efficient estimation of word representations in vector space. CoRR. abs/1301.3781, (2013).
18. *Mikolov, T. et al.:* Distributed representations of words and phrases and their compositionality. In: Burges, C. J. C. et al. (eds.) Advances in neural information processing systems 26. pp. 3111–3119 Curran Associates, Inc. (2013).
19. *Nivre, J. et al.:* Universal dependencies v1: A multilingual treebank collection. In: Proceedings of the tenth international conference on language resources and evaluation (lrec'16). pp. 1659–1666 (2016).
20. *Nivre, J. et al.:* Maltparser: A data-driven parser-generator for dependency parsing. In: LREC. pp. 2216–2219 (2006).
21. *Olga, L., Tatiana, S.:* GramEval 2020 Shared Task: Russian Full Morphology and Dependency Parsing. In: Computational linguistics and intellectual technologies: Papers from the annual conference "Dialogue". (2020).
22. *Peters, M. E. et al.:* Deep contextualized word representations. CoRR. abs/1802.05365, (2018).
23. *Sorokin, A. et al.:* MorphoRuEval-2017: An evaluation track for the automatic morphological analysis methods for russian. 2, 297–313 (2017).
24. *Zhou, J., Zhao, H.:* Head-driven phrase structure grammar parsing on Penn treebank. In: Proceedings of the 57th annual meeting of the association for computational linguistics. pp. 2396–2408 Association for Computational Linguistics, Florence, Italy (2019).