BERT implementation for detecting adverse drug effects mentions in Russian

Andrey Gusev* **Anna Kuznetsova*** **Anna Polyanskaya*** **Egor Yatsishin***
National Research University Higher School of Economics, Moscow
{aagusev_2,adkuznetsova_3,akpolyanskaya,esyatsishin}@edu.hse.ru

Abstract

This paper describes a system developed for the Social Media Mining for Health (SMM4H) 2020 shared task. Our team participated in the second subtask for Russian language creating a system to detect adverse drug reaction (ADR) presence in a text. For our submission, we exploited an ensemble model architecture, combining BERT's extension for Russian language, Logistic Regression and domain-specific preprocessing pipeline. Our system was ranked first among others, achieving F-score of 0.51. We have made our code publicly available[1].

1 Introduction

In this paper, we focus on the problem of discovering the presence of adverse drug reaction (ADR) concepts in twitter posts as part of the The Social Media Mining for Health Applications (SMM4H) Shared Task (Klein et al., 2020). The paper is based on the participation of our team in the Russian language segment of the second task: ADR presence classification. Organizers of SMM4H 2020 Task 2 provided datasets of Russian tweets with binary annotation indicating the presence or absence of ADRs in each post. The aim of the task was to develop a system to classify the tweets according to the presence of ADRs. Texts were given in a raw form, so they contained misspellings, slang, emojis, hashtags, usernames and were quite noisy. This year is the first time for a distinct set of Russian tweets to be included in the task. We tested and compared several different approaches for solving such type of classification task, including classical Machine Learning approaches and neural networks, and also different preprocessing pipelines.

2 Data

2.1 Datasets

The main dataset consists of the training set (7,612 tweets), validation set (1,522 tweets) and test set (1,903 tweets). The dataset is highly imbalanced with only 666 tweets labeled as mentioning ADR (hardly 9%). We used several techniques to overcome this problem, one of them being an attempt to create some additional data. We used RuDReC corpus[2] (Tutubalina et al., 2020) and manually labeled about 1,800 drug reviews (627 positive and 1195 negative).

It would be wrong to assume that these drug reviews are completely identical to the tweets from the main set in terms of linguistic features, so we did a simple analysis, which gave us several insights. First of all, the main lexicon of this two types of texts is quite similar, with the only two substantial differences:

1. Reviews tend to mention the cost or, to be more specific, the expensiveness of a drug much more often than tweets do, leading to the higher distribution of words like *expensive* and *overpriced*;

* Equal contribution.

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

[1] https://github.com/toskn/nru_hse_team_hlpl2020

[2] https://github.com/cimm-kzn/RuDReC

Proceedings of the 5th Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task, pages 46–50
Barcelona, Spain (Online), December 12, 2020.

2. In general, reviewers' language is significantly more grammatically correct and uses less slang and word shortenings.

The results of using this extended dataset are described in Section 4.

2.2 Preprocessing

Our approach for text preprocessing is to some extent based on the one used in (Ellendorff et al., 2019). The following changes were made using Python programming language:

- Tokenization using CrazyTokenizer from RedditScore package[3] based on spaCy tokenizer for Russian[4];
- Lowercasing, except all-caps words such as АД "antidepressants";
- All ё replaced with е;
- Urls replaced with ЮРЛ "url" placeholder;
- Usernames replaced with ЮЗЕРНЕЙМ "username" placeholder;
- Hashtags replaced with ХЕШТЕГ "hashtag" placeholder;
- Numbers replaced with NUM placeholder;
- Measures such as кг "kg" and мл "ml" replaced with MEASURE placeholder;
- Emojis replaced with POS_EMOJI for ones with positive meanings, NEG_EMOJI for ones with negative meanings and NEUTRAL_EMOJI for ones related to health issues as they could be semantically important;
- 3 or more repetitive letters normalized to 1;
- Line breaks deleted;
- Stopwords deleted. We used NLTK stopword list for Russian extended with some slang words as кароч "well", типа "like", прост "just", etc.;

Then we applied the following procedures, creating individual datasets for various combinations of them:

1. Leaving or deleting punctuation as tokens;
2. Lemmatization using PyMorphy2 (Korobov, 2015);
3. Stemming using Snowball Stemmer for Russian (Porter, 2001) via NLTK (Bird et al., 2009);

We chose PyMorphy2 over Mystem (Segalovich, 2003) for several reasons, first of them being the time, required to process the data on our OS (Windows). We also prefered pymorphy's lemmatization of unknown words (mostly, names of drugs and medical terms).

3 Method

3.1 Machine Learning

At first, classical Machine Learning models were trained to classify posts on original data without preprocessing. The data was represented as TF-IDF vectors. We decided to proceed with Support Vector Machine with simple linear kernel (LinearSVM), Logistic Regression model (LogReg) and Gradient Boosting Machine (GBM).

3.2 Deep Learning

For this approach, we explored the implementation of Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018): its extensions for Russian language — RuBert (Kuratov and Arkhipov, 2019) and Conversational RuBERT from DeepPavlov framework (Burtsev et al., 2018). RuBERT had the following characteristics: cased, 12-layer, 768-hidden, 12-heads, 180M parameters, and was fine-tuned with initialization from multilingual BERT on the Russian part of Wikipedia and news data. Conversational RuBERT had the same characteristics and was in turn fine-tuned with RuBERT on OpenSubtitles (Lison and Tiedemann, 2016). Due to the imbalance of classes, as mentioned above — 9% positive to 91% negative, we used following models:

[3] https://github.com/crazyfrogspb/RedditScore

[4] https://github.com/aatimofeev/spacy_russian_tokenizer

Name	Base model	Additions
RuBERT	RuBERT	—
Conv	Conversational RuBERT	—
ConvUnder	Conversational RuBERT	Undersampling
ConvLogReg	Conversational RuBERT	Undersampling + LogReg

Table 1: Model architectures.

For Undersampling approach we split negative class into N equal-sized folds, and combined each split with positive samples, giving that a share of negative samples is almost 0.5. Then we trained N models and stacked their probabilistic predictions for constructing an ensemble architecture. At first, we applied the majority voting method in order to get final answers. Beside that, Logistic Regression model was also used as the ensemble combiner.

4 Experiments

We compared classical ML models with basic BERT models on the original data without any preprocessing. The results reached by classical ML models, which can be found in Table 2, were not competitive, thus we didn't proceed with this approach.

Model	Validation F-score
LinearSVM	0.28
LogReg	0.07
GBM	0.29

Table 2: Classical ML models' results on original data

BERT models showed an increase in F-score, with Conversational RuBERT being slightly ahead of RuBERT. After deciding to stick with the Conversational RuBERT model, we made cross validation in search of the optimal parameters. Further experiments were conducted with Conversational RuBERT model with batch size equal to 32, dropout probability for non-Bert layers 0.4 and learning rate 10^{-5}. All results of testing can be seen in Table 3.

Considering using variants of dataset with additional texts, results were not promising, as there were no visible improvements and the F-score even decreased by 0.05. We believe that the reason for such behaviour lies in the dissimilarities of two types of texts being presumably more significant than we described in Subection 2.1. Implementation of the ensemble architecture proved to be successful and brought up an increase in F-score by 0.05 compared to non-ensemble models when using the ConvLogReg model. It should be noted here that we used a relatively small number of training epochs for models with undersampling, due to the high risk of overfitting.

name	n. models	batch size	n. epochs	last epoch	best epoch
RuBERT	1	64	7	0.36	0.37
Conv	1	64	7	0.45	0.47
ConvUnder	5	32	3	0.45	0.49
			4	0.43	
			7	0.45	
		64	4	0.42	0.46
ConvLogReg	5	64	4	0.49	0.49
	6	32	2	**0.51**	**0.51**
			4	0.48	

Table 3: F-score on validation set for different model configurations.

Further experiments were conducted in order to evaluate which pipeline of data preprocessing suits this task the most. Judging by the results shown in Table 4, simple preprocessing without lemmatization, stemming or punctuation deletion works the best, while any other type of preprocessing leads to a decrease in F-score. We discuss the reasons for that in Subsection 5.1.

prep	del punct	lem	stem	last epoch		best epoch	
				RuBERT	Conv	RuBERT	Conv
no				0.40	0.42	0.41	0.45
yes	no	no	no	0.36	**0.45**	0.37	**0.47**
		yes		0.26	0.34	0.26	0.40
			yes	0.29	0.34	0.29	0.39
	yes	no	no	0.37	0.36	0.38	0.43
		yes		0.33	0.35	0.33	0.38
			yes	0.30	0.37	0.32	0.38

Table 4: F-scores of simple models trained on main data with different preprocessing applied both to training and validation sets.

Given the results above, we settled on using ConvLogReg with 6 splits and small number of epochs. We also submitted one Conv model for comparison. Scores for the final models can be found in Table 5.

model		scores			
		Validation			Test
name	n epochs	precision	recall	F-score	
Conv	6	0.39	0.52	0.45	0.48
ConvLogReg	2	**0.51**	0.45	**0.51**	**0.51**
	4	0.45	**0.58**	0.48	0.50

Table 5: Metrics for three final submissions on validation and test sets.

5 Conclusion

In this work, we have explored an application of Bidirectional Encoder Representations from Transformers (BERT) to the task of text classification in Russian. We have empirically evaluated different versions of tuned RuBERT and preprocessing pipelines against F-score for the "positive" class and experiments have shown that logistic regression trained on the result of a six Conversational RuBERT models ensemble trained on the undersampled data with light preprocessing and tokenized punctuation outperforms every other model, providing a new baseline for ADR presence classification in Russian with F-score 0.51, precision 0.45 and recall 0.60 on the test data.

5.1 Discussion

Our research showed that stemming, lemmatization and punctuation removal when working with Russian language only decreases the final scores. Russian is different from English in such a way that the word order in Russian is free to some extent and syntactic relations are mostly encoded by morphology and punctuation. Knowing that BERT is capable of capturing hierarchy-sensitive and syntactic dependencies (Goldberg, 2019), it becomes obvious, that when dependencies indicators are blurred or removed the results worsen. In addition, some punctuation has its own semantics. For example, "(" means "sad" and "!!!!" can mean high importance. We don't see any premises to remove such kind of data from the dataset. Another point of discussion is the presence of mistakes in the datasets. In the training data we have found some debatable annotations and some which are erroneous for sure. These mistakes could possibly affect the performance of our model.

5.2 Future work

One potential objective for future improvement is to implement rule-based approach to the task, as mixed systems are known to perform better (Ray and Chakrabarti, 2019). We have already made some advancements on this path, but there is still a lot of research to perform. Also, we hope to continue enhancing the results by extending the dataset from Russian social networks and RuDReC corpus.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. 01.

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nikolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yurii Kuratov, Denis Kuznetsov, Alexey Litinsky, Varvara Logacheva, Alexey Lymar, Valentin Malykh, Maxim Petrov, Vadim Polulyakh, Leonid Pugachev, Alexey Sorokin, Maria Vikhreva, and Marat Zaynutdinov. 2018. Deeppavlov: Open-source library for dialogue systems. 07.

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

Tilia Ellendorff, Lenz Furrer, Nicola Colic, Noëmi Aepli, and Fabio Rinaldi. 2019. Approaching SMM4H with merged models and multi-task learning. In *Proceedings of the Fourth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*, pages 58–61, Florence, Italy, August. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287.

Ari Z. Klein, Alimova Ilseyar, Ivan Flores, Arjun Magge, Zulfat Miftahutdinov, Anne-Lyse Minard, Karen O'Connor, Abeed Sarker, Elena Tutubalina, Davy Weissenbacher, and Graciela Gonzalez-Hernandez. 2020. Overview of the fifth social media mining for health applications (#smm4h) shared tasks at coling 2020. *In Proceedings of the Fifth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*.

Mikhail Korobov. 2015. Morphological analyzer and generator for russian and ukrainian languages. In Mikhail Yu. Khachay, Natalia Konstantinova, Alexander Panchenko, Dmitry I. Ignatov, and Valeri G. Labunets, editors, *Analysis of Images, Social Networks and Texts*, volume 542 of *Communications in Computer and Information Science*, pages 320–332. Springer International Publishing.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *CoRR*, abs/1905.07213.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Martin F. Porter. 2001. Snowball: A language for stemming algorithms. Published online, October. Accessed 11.03.2008, 15.00h.

Paramita Ray and Amlan Chakrabarti. 2019. A mixed approach of deep learning method and rule-based method to improve aspect level sentiment analysis. *Applied Computing and Informatics*.

Ilya Segalovich. 2003. A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. In Hamid R. Arabnia and Elena B. Kozerenko, editors, *MLMTA*, pages 273–280. CSREA Press.

Elena Tutubalina, Ilseyar Alimova, Zulfat Miftahutdinov, Andrey Sakhovskiy, Valentin Malykh, and Sergey Nikolenko. 2020. The russian drug reaction corpus and neural models for drug reactions and effectiveness detection in user reviews. *Bioinformatics*, 07. btaa675.