Some Features of Sentiment Analysis for Russian Language Posts and Comments from Social Networks

N Sidorov and S Slastnikov
National Research University Higher School of Economics

Abstract. Sentiment analysis of different language texts is one of the very popular machine learning tasks. The complexity of its solution depends both on the characteristics of a particular language, and on the length of the evaluated texts. In our work, we consider the task of creating a sentiment analysis software tool for Russian posts and comments from the most popular social networks without any domain restriction. The features of constructing both the algorithmic and the software parts of the problem are described, some quality and performance metrics of the suggested neural network system are presented.

Keywords—sentiment analyses, embedding, convolutional neural network, the Russian language.

1. Introduction
With the rapid development of the Internet, people tend to use blogs, social networks or other web sites to publicize their attitude to various events or objects. This information could be beneficial for different organizations or even governments if they need to analyze people’s reaction to something. The difficulty in conducting such analyses is that in order to provide some meaningful statistics on public opinion, one has to collect and examine thousands of posts and comments. It seems impossible to cope with this task in a short time manually. However, in recent years, machine-learning techniques have achieved remarkable results dealing with large amounts of data. In particular, there are algorithms that allow users to solve some natural language processing problems, such as text classification or sentiment analyses.

Nevertheless, despite the relatively high quality of these methods, they are far from being excellent, which provides a huge area for improvements. Moreover, due to a drastically different language system of Russian, many models that perform well for English are not appropriate in all aspects to Russian. That is why a number of problems in Russian still do not have well-established solutions.

The aim of this study is to develop a system for sentiment identification of texts from social networks. To that end, some methods of natural language processing were analyzed and applied to the available data. This paper describes different model variants, compares them and suggests its own version of the system structure for processing posts and comments.

The proposed system predicts one of six classes for each given text. Five of these classes were introduced in [1]: “negative”, “neutral”, “positive”, “skip” and “speech”. The present study also adds a “humor” class as a sixth one.

The classification framework will be a part of an application for social networks data mining and analysis. Those who want to find posts in social networks on the required topic and receive some statistics on public opinion could use this application for their purposes.

The problems of automatic natural language processing in general and, in particular, texts classification have been discussed extensively in recent years. For example, in [2], the authors design a...
tree representation of tweets and introduce several part-of-speech-specific prior polarity features in order to conduct sentiment analyses of Twitter data.

Paper [3] demonstrates a neural network architecture that could be utilized in different tasks: part-of-speech tagging, named entity recognition, semantic role labeling, to name but a few. The main idea is to represent each word as a vector and, consequently, a sentence as a matrix. This matrix then goes as an input layer in convolutional neural network. The lookup table between words and vectors modifies while being trained. In [4], the authors use a slight variant of model [3] and pre-trained Word2Vec embeddings (vector representation of words). They test the model in several sentence classification benchmarks, showing excellent results and concluding that it is possible to improve the quality of a model by replacing the randomly initialized lookup table with pre-trained embeddings. More recent research [5] analyses the influence of variance in hyperparameters in the same convolutional model on different datasets and proposes the best variant of settings.

So far, all the described studies have dealt with the English language. Apparently, application of the proposed algorithms to Russian requires a huge dataset in Russian. Such a dataset, called RuSentiment, is presented in [1]. It contains more than 30000 Russian posts from VKontakte, distributed in 5 classes: “negative”, “neutral”, “positive”, “skip” and “speech”.

Paper [6] describes ELMo embeddings – a more sophisticated vector representation for natural words. This type of embeddings trained by DeepPavlov [7] for Russian words from Twitter is free to use.

There already exist some works on sentiment analyses of posts from Russian social networks. For instance, in [8], the authors focus on the accounts of Russian politicians. They also utilize RuSentiment and apply bidirectional recurrent neural network model. Another research with recursive neural network is presented in [9]. In addition, paper [10] studies the level of social tension by applying support vector machine to some data from VKontakte social network.

2. Datasets
This section describes the data used for training and testing the system.

Since social media posts and comments can vary in style and structure, this study relies on several different datasets to train and test the models.

1) **RuSentiment.** It contains 31185 posts in Russian from VKontakte. All the posts are divided into 5 classes:
- “Negative” – explicit and implicit forms of negative expressions.
- “Neutral” – texts without sentiment.
- “Positive” – explicit and implicit forms of positive expressions.
- “Skip” – unclear posts.
- “Speech” – greetings and congratulations.

Figure 1 presents the distribution of posts between these classes.

![Figure 1. The distribution of objects in RuSentiment.](image1)

For more detailed information on RuSentiment dataset, please refer to the original article [5].

2) **Uniform RuSentiment.** However, as one can see from the histogram for RuSentiment, there are too many neutral posts in this dataset. In order to check whether the models are overlearned.  

![Figure 2. The distribution of objects in News dataset.](image2)
for this class, the present study constructs a Unifrom RuSentiment dataset by leaving exactly 3000 objects in each class.

3) **News.** Moreover, RuSentiment mostly contains rather short expressions, such as “Ураа, нам сегодня три года…😊”, while some users may be interested in more informative, news-like, posts. That is why the system should cope with sentiment detection in both cases: with long formal posts and short comments. To that end, this study also utilizes news’ sentiment dataset from the Kaggle competition [11] that includes 8263 news assigned to 3 classes: “negative”, “neutral” and “positive”.

4) **FUN.** Given a large number of jokes on the Internet, this study also introduces the “humor” class. So, the system should recognize, if a post is a joke or not. The FUN dataset [12], which consists of 312877 short texts, is used for training. Approximately a half of these texts are humorous, while the others are non-humorous.

5) **Pikabu.** This study also creates a sample of 7301 posts from the Pikabu entertainment community, containing positive and negative texts.

**Figure 3.** The distribution of objects in Pikabu dataset.

### 3. Methods and experiments

#### 3.1. **TF-IDF vectorisation**

One of the simplest techniques to convert texts into vectors is a TF-IDF vectorization. It is based on the comparison between word’s frequency in the given text (document) and its frequency in other texts. Each document is then represented as a vector of the same dimensionality as the number of words in the vocabulary. The coordinate with number \( i \) corresponds to the \( i \)-th word in the vocabulary and is calculated as the TF-IDF measure for corresponding word and given text. Various machine-learning algorithms can subsequently use such vector representation as feature description of documents.

This study performs the TF-IDF vectorization for texts from RuSentiment dataset and applies then random forest classifier and SVM (support-vector machine) classifier. The hyperparameters for these algorithms were chosen with 5-folds cross-validation. Common metrics, such as accuracy, precision, recall and F1-score were calculated in order to measure the quality of predictions. The results of predictions on test dataset are the following:

**Table 1.** The results of classification by Random Forest and SVM with TF-IDF vectorization.

| Method    | Random Forest | SVM          |
|-----------|---------------|--------------|
| **Best hyperparameters** | 500 estimators, 100 – max depth | Linear kernel, \( C = 1 \) |
| **Metrics** | Precision: 0.85  Recall: 0.43  F1: 0.44  Accuracy: 0.64 | Precision: 0.68  Recall: 0.59  F1: 0.61  Accuracy: 0.70 |

The results show that such models are better than random guessing. Nevertheless, they are not convincing enough. A possible reason for such unsatisfactory results is that the described approach has several drawbacks. First, the TF-IDF vectorization does not take into account the context of words, as it only counts the frequencies. Secondly, the vocabulary is constrained by the words in the training set. Posts in social networks are not linked to one topic, so many of them would probably contain a significant amount of word out of trained vocabulary. Moreover, adding new words to the vocabulary increases the dimensionality of vectors, which leads to the growth in required memory and computational time. Therefore, this study concludes that TF-IDF is not appropriate for the task.
3.2. Pre-trained embeddings

In [4], the author claims that pre-trained embeddings are useful for many NLP tasks. Let us, as usual, refer to the word “embedding” as a vector representation of a word, constructed in such a way that semantically close words have close representations in the vector space. There are different kinds of embeddings: Word2Vec, FastText, Elmo, to name but a few. Such models for conversion are trained on a large number of texts, without being linked to a specific task. This study exploits the ELMo embeddings [6], as they are the most advanced ones, constructed by a recurrent neural network considering the context of a word. Particularly, this paper use ELMo embeddings, pre-trained on millions of words from Russian Twitter by DeepPavlov.

3.3. CNN

After turning each word into vector one can represent the whole text as a matrix and treat it as an image. Accordingly, it becomes possible to apply models that usually work with images to the text. In this project, the convolutional neural network (CNN) is chosen as the classification model. This type of neural networks has several filter matrices, and the task of these filters is to recognize different patterns in the considered object. When working with texts, filters have the same width as the dimension of the vector representation of each word, while the height of the filter matrices can vary, so the filter with height $n$ will try to find a pattern in $n$-words collocations. One can use several filters with different heights in order to detect various patterns. The scalar product between different parts of the text and each filter is calculated, after which the obtained numbers pass to a fully connected layer, where they are converted to the probabilities that the object belongs to a particular class.

The CNN model with such architecture achieves rather high quality on the test dataset. However, practice shows that the model too often predicts the “neutral” class. This may happen because the distribution of classes in the train dataset is not uniform, so, the number of “neutral” texts outweigh the number of texts in other classes.

There exist several ways to overcome this obstacle. One of them is to align classes by removing some objects. This study leaves equal amount of texts in all classes in RuSentiment, constructing Uniform RuSentiment dataset. As expected, the predictions become more uniform but the quality on the original dataset falls because many “neutral” posts are now assigned to other classes.

Another possible reason for such a significant number of “neutral” texts in practice is that many posts on the Internet do not express someone’s emotions but contain news information that could also be positive or negative by itself. The model trained on RuSentiment does not recognize the sentiment of such posts because they differ from examples presented in RuSentiment. Therefore, a dataset containing approximately 8000 news texts was found. It was used to train models to recognize color in news.

Another idea of this study is to add a “humor” class, as the jokes in the social networks are ubiquitous. A part of FUN dataset [8] is used for humor recognition.

Thus, at this stage, the study considered 5 datasets: RuSentiment, Uniform RuSentiment, FUN, Combined RuSentiment and FUN, News. They were divided into training and test samples. A CNN model was trained on each of the training samples and tested on some of the test samples. The results of the experiments are presented in Table 2.

| Model/Dataset | Dataset | ruSentiment Precision: 0.76 Recall: 0.74 F1: 0.75 Accuracy: 0.79 | Uniform RuSentiment Precision: 0.78 Recall: 0.74 F1: 0.74 Accuracy: 0.74 | Humor --- | Combined RuSentiment and Humor --- | News Precision: 0.11 Recall: 0.20 F1: 0.13 Accuracy: 0.49 |
|---------------|---------|---------------------------------------------------------------|---------------------------------------------------------------|---------|---------------------------------------------------------------|---------|
| 5-class, trained on RuSentiment | Precision: 0.66 Recall: 0.75 F1: 0.69 Accuracy: 0.70 | Precision: 0.75 Recall: 0.76 F1: 0.75 Accuracy: 0.76 | --- | --- | Precision: 0.12 Recall: 0.14 F1: 0.09 Accuracy: 0.23 |

Table 2. The quality of classification by models, trained on different datasets.
The results showed that if the model was trained and tested on samples from the same dataset, its quality is rather high (more than 0.7). However, the model trained on short texts gives poor results when applied to news texts, and vice versa. Thus, these models are not interchangeable.

3.4. The system of models
We can conclude that in order to correctly determine the sentiment of any text from social networks, one first needs to identify the type of text: news or comment, and then build the appropriate embeddings and apply the corresponding model for classification. To that end, a simple and fast model (LGBM Classifier) was trained to classify texts by their type after preliminary TF-IDF vectorization. It is much easier to determine the type of text than the sentiment, so such simple methods show very high results (the quality is higher than 0.85).

So, the proposed system has the following structure:

Moreover, it was decided to expand the "Positive" and "Negative" classes by adding posts from the Pikabu entertainment community. Each Pikabu post has a set of tags by which it is possible to determine its sentiment. A function for parsing posts was implemented and new training and test samples were formed. Since posts with Pikabu mostly have rather clear sentiment, they were used for retraining the comments-model.

In addition, to reduce the number of neutral predictions of the news model, it was retrained by selecting weights for each class to balance them. Thus, the model is penalized more if it recognizes a positive or negative text as neutral, and less if it recognizes a neutral text as positive or negative.

Table 3. The quality of the final system with pre-separation of the posts.

|     | Precision | Recall | F1-score |
|-----|-----------|--------|----------|
| Humor | 0.86 | 0.73 | 0.79 |
| Negative | 0.67 | 0.63 | 0.65 |
| Neutral | 0.71 | 0.8 | 0.76 |
| Positive | 0.71 | 0.69 | 0.7 |
| Skip | 0.52 | 0.44 | 0.48 |
| Speech | 0.91 | 0.95 | 0.93 |
| Macro AVG | 0.73 | 0.71 | 0.72 |
| Weighted AVG | 0.74 | 0.74 | 0.73 |
| Accuracy | | | 0.74 |

Table 4. The quality of one model in all test samples.

|     | Precision | Recall | F1-score |
|-----|-----------|--------|----------|
| Humor | 0 | 0 | 0 |
| Negative | 0.57 | 0.14 | 0.23 |
| Neutral | 0.37 | 0.92 | 0.53 |
| Positive | 0.71 | 0.20 | 0.31 |
| Skip | 0.34 | 0.49 | 0.4 |
| Speech | 0.87 | 0.9 | 0.88 |
| Macro AVG | 0.48 | 0.44 | 0.39 |
| Weighted AVG | 0.49 | 0.43 | 0.36 |
| Accuracy | | | 0.43 |
The quality of the final system is shown in Table 3 in comparison with the quality of one model trained on RuSentiment (Table 4).

4. Conclusion

This paper has proposed a system for classifying Russian-language texts from social networks by their sentiment basing on convolutional neural networks. The average quality of the resulting system on diverse data sets is 74%.

Original features of the proposed system:
- classification into six classes, including "humor", while most similar works focus on three classes
- the system classifies texts without being linked to any topic area, which makes it stand out from its analogues that work, for example, only with movie reviews or restaurant reviews
- a new set of data from Pikabu was generated to expand the system's capabilities

There are various prospects for future work: try systems with a different structure, for example, build a binary model for each class; find ways to expand the training data sets; apply models with a different architecture; train the system to detect sarcasm, etc.

5. References

[1] Rogers A, Romanov A, Rumshisky A, Volkova S, Gronas M and Gribov A 2018 RuSentiment: An enriched sentiment analysis dataset for social media in Russian Proc. Int. Conf. on computational linguistics pp 755–63
[2] Agarwal A, Xie B, Vovsha I, Rambow O and Passonneau, R J 2011 Sentiment analysis of twitter data Proceedings of the workshop on language in social media (LSM 2011) pp 30–8
[3] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K and Kuksa P 2011 Natural language processing (almost) from scratch Journal of machine learning research pp 2493–537
[4] Kim Y 2014 Convolutional neural networks for sentence classification arXiv (Preprint 1408.5882)
[5] Zhang Y and Wallace B A 2015 sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification arXiv (Preprint 1510.03820)
[6] Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K and Zettlemoyer L 2018 Deep contextualized word representations. arXiv (Preprint 1802.05365)
[7] DeepPavlov https://deeppavlov.readthedocs.io/en/0.2.0/index.html (2020)
[8] Svetlov K and Platonov K 2019 Sentiment Analysis of Posts and Comments in the Accounts of Russian Politicians on the Social Network 25th Conf. of Open Innovations Association (FRUCT) Vol 15, pp 583
[9] Averchenkov AV, Budylskii D, Podvesovskii A, Averchenkov A, Rytov M and Yakimov A 2015 Hierarchical deep learning: a promising technique for opinion monitoring and sentiment analysis in Russian-language social networks InCreativity in Intelligent Technologies and Data Science: First Conference, CIT&DS Vol 15, p 583
[10] Donchenko D, Ovchar N, Sadovnikova N, Parygin D, Shabalina O and Ather D 2017 Analysis of comments of users of social networks to assess the level of social tension Procedia Computer Science pp 359-67.
[11] Kaggle https://www.kaggle.com/c/sentiment-analysis-in-russian/overview (2020)
[12] Blinov V, Bolotova-Baranova V and Braslavski P 2019 Large dataset and language model fine-tuning for humor recognition Proc. of the 57th annual meeting of the association for computational linguistics pp 4027–32