A Comparison of Machine Learning Methods of Sentiment Analysis Based on Russian Language Twitter Data

Andrey Zvonarev[0000−0003−2191−2167] vodogrey@niuitmo.ru and
Andrey Bilyi[0000−0002−6133−4368] bilyi_andrei@mail.ru

ITMO University, 49 Kronverksky Pr., St. Petersburg, 197101, Russia

Abstract. The paper is focused on comparing the performance of different techniques of text tonality analysis, a widely used approach in business to conduct social listening research. However, there are still debates on what type of models perform better for NLP classification tasks. On a corpus of Russian language tweets three models were tested to solve binary classification problem: Logistic regression (LR), XGBoost classifier and Convolutional Neural Network (CNN). The paper descriptively overviews main techniques useful for data cleaning and preprocessing for these methods and covers its possible fitfalls. Based on the study CNN showed best results among chosen models, which goes in line with several articles in the field for other than Russian languages. Together with high predictive results, neural networks exhibit a computational drawback - their performance is poor in terms of timings. Besides this, all methods showed high sensitivity to the way data is preprocessed, which is a product of Russian language variability. This leads to a conclusion that there is still a room for improvement - in future research more emphasis will be put on hyper parameter tuning for ”boosting type” models and extending list of applied methods.

Keywords: text tonality · sentiment analysis · logistic regression · CNN · XGBoost · natural language processing

1 Field overview and prediction task description

Nowadays, more and more communication, services and goods transfer to the Internet where the information is basically provided in the form of text. In this regard, the task of determining the emotional state of a person without personal communication is becoming increasingly crucial. The perspectives of this field are that, based on the textual information, it is possible to determine a mood of a person, or to estimate a success of political or economic reforms, or to check person’s reaction to particular event or decision.

Due to in practice it is not possible to identify the emotional coloring of large number of texts available in the Internet manually and, in addition, it is too labor-intensive, one of the possible ways to solve this problem is to use machine learning algorithms.

At the moment, there are several key methodologies for determining emotional coloring of the text:

- **Analysis using pre-complied dictionary**[1]. Such dictionaries consist of pre-prepared template words, phrases and their combinations with emotional coloring characteristic of each element. Moreover, for determining emotional coloring with improved accuracy of tonality assessment corpus linguistics can be used[2]. Nevertheless, in paper[3], the authors encountered differences in the expression of emotions in English and Russian languages when using a bilingual corpus. The positive experience of translation of dictionaries is presented in the paper[4], where the authors translated the dictionary of emotionally colored vocabulary in English into Chinese. The assessment is made on set of positive and negative patterns found. With the explicit allocation of one of them to the text or passage, the class that scored more points is set. If there is no obvious predominance, the rating is set to neutral. The main disadvantage is the procedure for compiling glossaries of terms indicating the weight of phrases. Also, these dictionaries must be prepared for a specific area.
- **Analysis with the use of machine learning methods** has recently become the most widespread because of reducing the influence factor of human impact on the assessment. In comparison with pre-complied dictionary method where the assessment is set by person, the assessment in machine learning method is set by independently identified patterns in the text even with achieving recognizing sarcasm and irony.

This paper compares several machine learning techniques to solve binary classification problem of text tonality analysis. The dataset used for this task is a corpus of Russian language tweets. After several data cleaning stages logistic regression (LR), XGBoost classifier and convolutional neural network (CNN) are applied. The study compares those models in terms of accuracy and F1 score and discusses possible pros and cons for these methods.

2 Data preprocessing and models overview

Proposed methods are based on the use of machine learning methods. The developed prototype allows to assess texts as ‘positively’ or ‘negatively’ coloured. To train these models a set of short texts were collected. Using RuTweetCorp[5] dataset, which is a unique source of Russian-language tweets, collected via twitter API interface. The dataset is valuable for being a single open source collection of Russian language texts on the market with predefined texts tonality. RuTweetCorp’s database allows Russian researchers to test modern modelling approaches of NLP on their native language corpus. The dataset of 225 000 tweets[2] was randomly split to the training set (70%) and test set (30%).

2.1 Data cleaning and preparation

Before applying discussed above methods, data cleaning stage should be completed. To prepare the dataset for the models I used in the paper several classical for the industry approaches are applied:

- **Capitalisation**. The purpose of this stage is to make a proper text cleaning and frequency calculation. Actually it makes no difference what type of register level to use for the modelling since, mathematically, models' algorithms transform words into digits, thus any case type can be chosen. In the current paper I have used lowercase register;
- **Punctuation cleaning**. As a part of data preprocessing researchers usually drop all punctuation marks, digits, links, etc. from texts since in most of the times it does not give any impact on emotional meaning of tweets. Another effect of these symbols on the analysis is irrelevant results in terms of word frequencies and affinities. Punctuation marks are comparatively often appear in sentences which leads to extremely high metrics for them while in reality these symbols almost always make no impact on text tonality;
- **Lemmatisation**. Other well-known in the industry approaches to reduce the number of words carrying similar emotional meanings are lemmatisation and stemming. Both of these methods transform words from its full form into their short parent version. The difference between methods is that lemmatisation derives an infinitive form of the words while stemming simply cuts beginnings and endings of words to obtain a root. In the current paper I use lemmatisation as I believe it to be more efficient[6];
- **Stop-words deletion**. Next step making data less noisy is dropping so-called stop-words from the dataset. "Stop-words"[7] are words that extremely frequently appear in text but make no impact on text meanings. Classical examples of such words are articles (such as the, is, at, which, and on). Another type of such words may be swear words, which may have one infinitive form but tens of word variations with both positive and negative meanings.

Besides well-known in literature methods of texts cleaning in current paper I have followed additional step improving algorithm performance - concatenation of word "not" with the following one. It is intuitively understandable that the negation word "not" can significantly change the emotional meaning of a tweet. However, it was practically revealed that analysed models captured high correlation between "not"[8] word and negative target value, which afterwards resulted in high share of incorrect "negative" predictions. To overcome this issue I have concatenated the word "not" with the following one and used it as a new word, which led to performance growth.

[2] Out of which 'obviously positive' (114,911 records) and 'obviously negative' (111,923 records.) observations

Having done all that, tweets have been transformed to vectors consisting of cleaned infinitives of words. This still resulted in huge amount of unique words, which led to high memory and time consumption, while big chunk of words did not give any impact on model score. To optimise the modelling process words with frequency less than 3 were dropped. After that, a dictionary was created, which included remaining words to which a unique ID was assigned. Using such dictionary we have prepared two input datasets:

- **For LR and XGBoost**: a TF-IDF matrix (term frequency-inverse document frequency), which is a method of "Bag of words"(BoW) class. While term frequency denotes how many times a word is in a document (in one piece of text input), inverse document frequency calculates number of documents where a text has appeared and number of total documents in data. In this method all the words in the data are transformed into a list. Afterwards all the words in this list are assigned to each document as a vector. For example if there is a dataset of 5000 words, each document has a vector of 1 row and 5000 columns. Each column in this vector defines a word. If the word appears in the document, number of times the word is present in the text is assigned to the column corresponding to that word. When this process is done for all documents, illustration of data as per text frequency is obtained. Inverse Document Frequency (IDF) is calculated by using number of documents where a term has appeared and total number of documents in the data.
- **For CNN input**: a Document-Term Matrix of special form - rows of this matrix were corresponding to tweet ID, columns corresponded to the position of the word in a tweet, while in each cell of the matrix were written an ID of the word from dictionary. Such matrix may have large number of columns in case of presence of very long texts. This leads to high memory and time consumption during computation. To reduce the time, the matrix were cut to 23 columns, which significantly improved model calculation time on one side and left all relevant information on another (23 columns cover 99.65% of all tweet lengths)

2.2 Models description

For the purpose of the analysis we have tested several machine learning methods and compared its efficiency (more on this in section 3). We have compared three typical models in the text mining field to conduct semantic analysis of texts: logistic regression, XGboost and CNN. Short overview of these methods is provided below.

Logistic regression This is a well known approach used to solve binary classification problems.[9] Text mining is just one of many fields where logistic regression may be applied once the task can be transformed to binary classification type. Basically, the idea behind the method is to calculate the probability of tweet to be positive based on rules identified on a large set of data. Since the probability function (1) has a logistic form the model also got its name:

$$f(z) = \frac{1}{1 + e^{-x}} \tag{1}$$

where z is a set of model input factors (in our case these are vectors of TF-IDF matrix). More on this type of regression is given by Cramer[10].

XGBoost Another model we have decided to test on the dataset is XGBoost, which is highly credited by machine learning competitors[11]. This software nowadays exists for all popular data analysis languages and provides gradient boosting framework. XGBoost is used both for regression and classification problems, thus is expected to perform considerably well on binary classification problem and data we use in the current paper. More on the method is in original paper by Chen and Guestrin[12].

Convolutional Neural Network (CNN) The last but not least model type we have tested on the dataset is a CNN. Neural networks recently got high attention among data scientist due to ability to solve almost any type of problem once it is stated correctly. Text mining is one of the fields, where CNN showed high performance[13].

3 Results and discussion

To test the results of models performance we estimate shares of correct and incorrect predictions on train and test set. It is a practical and reliable way to deal with over- and under-fitting. To measure the learning performance of methods we use accuracy and F1-score metrics. These two metrices were chosen based on research in the literature for binary classification[14].

Below we demonstrate obtained results for all three models. Table 1 shows Logistic Regression results:

Table 1. Logistic Regression results

	Accuracy	F1	Time
Training set	84.7 %	84.9%	45.2s
Testing set	76.7%	76.9%	

Based on the results one can make a conclusion, that model is over-fitted. This is for sure not a good sign, however, it can be fine-tuned by changing hyper-parameters of the model and using cross-validation techniques. Overall, the score on test is considerably high. Another think to note - training of logistic regression take very small amount of time. This makes the method a good starting model to test how data preparation stages influence model performance.

XGBoost training took much more time. Results you can see in the table below.

Table 2. XGBoost results

	Accuracy	F1	Time
Training set	75.8%	74.6%	9h 44m 47s
Testing set	72.8%	71.3%	

This model shows worse results than logistic regression. Probably, it happened because it needs a better setup of hyper parameters. Since in literature XGBoost proved to be one of the best models for these type of tasks, we are motivated to pay more attention to work with this model in the future.

Table 3. Convolutional neural network results

	Accuracy	F1	Time
Training set	82.9%	84.2%	6h 11m 24s
Testing set	79.5%	78.1%	

Table 3 shows convolutional neural network results. CNN demonstrates the greatest performance on a test set, but you may also notice that training phase took a lot of time as well. For even better results another tokenizer algorithm may be used, such as n-grams and add more words and forms into thesaurus for better lemmatization.

4 Conclusion

The paper provides with results of applying different machine learning techniques to solve the text tonality binary classification task. Three model types were estimated: logistic regression, XGBoost classifier and convolutional neural network. Each of them are well-known models among data scientists to solve tasks of that type. However, not that much studies exist where those models are applied and compared on social networks data and Russian language. Results demonstrate that based on F1 measure CNN performs better. However, training such a model needs much more time than LR. Hence, depending on available time and computing power for modelling LR may be preferred. Also, it was a surprise that XGBoost classifier showed significantly lower result than other models, while the framework demonstrated high performance on many data science competitions. We suspect that more time should be spent to find optimal hyper parameters of the model.

For future research we are planning to extend the number of models tested - at least, LightGBM and Word2Vec models are interesting frameworks showing good results for this type of problems. Besides that, more focus will be allocated on optimal hyper parameters search, cross-validation techniques and stacking of models. These we believe will lead to even greater model performance.

References

1. Alexeeva S., Koltsova E., Koltcov S. Linis-crowd.org: A lexical resource for Russian sentiment analysis of social media [Linis-crowd.org: lexicheskij resurs dl'a analiza tonal'nosti sotsial'no-politicheskix tekstov], Computational Linguistics and computantional ontologies: Proceedings of the XVIII joint Conference "Internet and modern society (IMS-2015)" [Kompyuternaya lingvistika i vyichislitelnyie ontologii: sbornik nauchnyih statey. Trudyi XVIII ob'edinennoy konferentsii Internet i sovremennoe obschestvo (IMS-2015)], St. Peterburg, pp. 25–34.
2. Zagibalov, T., Belyatskaya, K., Carroll, J.: In Computational Approaches to Subjectivity and Sentiment Analysis, 2010. – . 67– 72.
3. Meng X. Lost in translations? building sentiment lexicons using context based machine translation / Meng X., Wei F.,etc. // COLING, 2012. – . 829–838.
4. Pazel'skaya, A., Solov'ev, A.: Metod opredeleniia emotsii v tekstakh na russkom yazike. Dialog-2011. Sb. Nauchnih statei / Vip. 11 (18).- .: RGGU, 2011.– .510-523.
5. Rubtsova, Y.: A method for development and analysis of short text corpus for the review classification task. In: Proceedings of Conferences Digital Libraries: Advanced Methods and Technologies, Digital Collections, RCDL 2013, pp. 269–275 (2013)
6. Stemming and Lemmatization: A Comparison of Retrieval Performances, `http://www.lnse.org/papers/134-I3007.pdf`. Last accessed 8 Nov 2019
7. Luhn, H. P.:Keyword-in-Context Index for Technical Literature (KWIC Index). American Documentation. 11 (4): 288–295. CiteSeerX 10.1.1.468.1425. https://doi.org/10.1002/asi.5090110403.
8. Sanjiv D., Chen M. Yahoo! for Amazon: Extracting market sentiment from stock message boards // Proceedings of the Asia Pacific finance association annual conference (APFA). — 2001.
9. Logit-analiz, . Last accessed 12 Oct 2019.
10. Cramer, J. S. (2002). The origins of logistic regression (PDF) (Technical report). 119. Tinbergen Institute. pp. 167–178. https://doi.org/10.2139/ssrn.360300.
11. Awesome XGBoost, `https://github.com/dmlc/xgboost/tree/master/demo#machine-learning-challenge-winning-solutions`. Last accessed 10 Nov 2019
12. Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). ACM, New York, NY, USA, 785-794. https://doi.org/10.1145/2939672.2939785
13. Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning (ICML '08). ACM, New York, NY, USA, 160-167. https://doi.org/10.1145/1390156.1390177
14. Sokolova, M., Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing Management, 45(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002