Comparison of traditional machine learning methods and Google services in identifying tonality on Russian texts

A Shalkabayuli, A Kairbekov and Y Amangeldi
Institute of Information and Computational Technologies, Kazakhstan

ardak.shalkar@gmail.com, abylay.kairbekov@gmail.com, yerbolat101@gmail.com

Abstract. Classification of text is very important and interesting topic for research due to number of unstructured data and especially in the form of news. The problem arises when classifying news that is written in language other than english. Due to poor support of research done in this field and general language complexity, it is very hard to identify correctly, tone and sentiment of a text. This problems can be avoided by applying various techniques at many stages of this model. Choosing proper technique can be very tricky and would affect efficiency of a model and can save time as well as working time. The aim of this paper is to compare traditional machine learning techniques with Google services by using Translate API and see how well they perform while working with Russian texts. On traditional machine learning techniques after applying pre-processing methods which included stemming, parsing and indexing and by using various different methods of embedding turns our unstructured data into a structured data.

1. Introduction
Classification of news in the form of sentiment analysis can be very helpful in many ways including identifying tone of the news websites and generally capturing the tone of a text. This paper focuses on comparing traditional machine learning algorithms like SVM, Naive-Bayes, Logistic Regression with Google services that would automatically identify sentiment of a text. The main goal of this research is to see how well those 2 different approaches differ and how well they perform on Russian texts. The organization of this paper is given below: Section 2 contains description of the dataset that was used in this research. Section 3 explains methods and techniques that were used during this research. Section 4 demonstrates Empirical Analysis of our methods and their results. Section 5 demonstrates results which was shown on Google Services. Section 6 concludes the paper.

2. Data Description
This dataset consists of 8000 records of news that were classified into 3 main classes which are: positive, negative and neutral. This dataset was collected by Alem research group from various Kazakhstan news websites. There are 1500 of negative tonal, 4000 neutral and 2500 positive news in that dataset. All news are in russian language and contains many named entities that are connected to Kazakhstan. News articles are not preprocessed in any ways and contains many unnecessary texts that are not related to article itself.
3. Methods
For traditional methods it uses TF-IDF [1] as an embedding. Before transforming this text into TF-IDF matrix it requires a lot of preprocessing. First of all, stopword removal process, in this case we used default nltk corpus for Russian stopwords. It do not have many words but contains basic stopwords for Russian language. Right after that we removed all digits and dates from text because they do not contribute to the meaning of the text anyhow. Along with digits we have removed punctuations which would shorten our matrix very much. As a final step of preprocessing we have applied pymorphy2 [2] library which can transform Russian words and find their stemmed version. This step would remove many versions of the same words that appear in the text. Furthermore, Named entities were removed by library for removing russian Named Entities [3].After preprocessing by using scikit-learn we apply TfidfVectorizer to transform our text into TF-IDF vectors. This particular model uses unigrams as well as bigrams in counting vectors. This resulted in 100,000 dimensional vector with 8000 records. Here we have used 5 algorithms to validate our model they are: Naive-Bayes [4], SVM [5], LinearRegression [6], Decision Tree [7] and Random Forest [8]. Also scikit-learn’s cross-validation is used to check this algorithms.

4. Empirical Analysis
Because the primary goal of this research is to compare two methods of finding sentiment analysis on Russian news, main focus would be to compare their accuracy and classification report and compare their productivity and effectiveness. First let’s look at TF-IDF. After converting raw texts into TF-IDF vectors with bigrams and seeing each classes correlated words and phrases these are the results:  
1. Unigrams:  
   a. negative:  
      i. suspect  
      ii. criminal  
      iii. pre-trial  
      iv. bribe  
      v. arrest  
   b. positive:  
      i. deal  
      ii. check(review)
iii. court
iv. apprehend
v. investigation

2. Bigrams:
   a. negative:
      i. measure of restraint
      ii. criminal case
      iii. pre-trial investigation
      iv. large bribe
      v. large size
   b. positive:
      i. social project
      ii. national economy
      iii. presidential club
      iv. cooperation
      v. national minister

Figure 2. Cross Validation Scores.

As you can see from this correlation grasped the meanings of tone of overall messages. This shows that we got enough data to classify them accordingly. To further analyze this dataset let’s look at their cross validation scores of those 5 algorithms:

Table 1. Cross-validation score for algorithms.

| Algorithms       | Accuracy  |
|------------------|-----------|
| DecisionTreeClassifier | 0.554761  |
| LinearSVC        | 0.698417  |
| LogisticRegression | 0.697085  |
| MultinomialNB    | 0.637540  |
| RandomForestClassifier | 0.488200  |
As you can see from Figure above, cross-validation scores clearly shows that SVM and Linear Regression works very well on this text. Naive Bayes algorithm performs somewhat average whereas Decision Tree based algorithms perform poorly on this text. This further strengthens the notion that Decision Tree based algorithms perform poorly on any texts. But Random Forest Classifier is weaker in text classification. By analyzing their actual and predicted values we can see where our algorithms are making misclassification. Because SVM and Linear Regression do have similar scores we will only cover LinearSVC.

As you can see from Figure 3 most misclassification in SVM are due to neutral texts being classified as positive and vice versa more than 200 test records are being classified that way. Furthermore, negative tonal texts are also being classified as neutral is very high, more than 100 records. So this means that it is easier to our classifier to separate negative and positive texts but way harder to distinct positive from neutral and negative from neutral. This is somehow logical considering we are using word frequencies to distinguish between them. Let’s look at the same results but using Naive Bayes this time:

![Figure 3. Scores of LinearSVC.](image1)

![Figure 4. Scores of Multinomial Naive Bayes.](image2)

Even though accuracies are similar Naive Bayes has different misclassification results. This time the problem is not neutral records being misclassified as positive or negative but rather positive and negative texts being classified as neutral. But their values are way higher than LinearSVC algorithm. This means that this algorithms clearly identifies neutral records as neutral but having problems when it comes across positive and negative values. This in a way is the opposite of LinearSVC. LinearSVC outperforms other algorithms so let’s look at it’s classification report:

| Class     | Precision | Recall | F-1 score |
|-----------|-----------|--------|-----------|
| Negative  | 0.70      | 0.65   | 0.67      |
| Positive  | 0.69      | 0.66   | 0.67      |
| Neutral   | 0.70      | 0.74   | 0.72      |
| Average/Total | 0.70  | 0.70   | 0.70      |

5. Google services
Google Cloud Platform – it’s a such cloud infrastructure for working with different kind of Google services. This platform provide user friendly management tools as well as data storage, virtual machines deployment, hosting, big data and machine learning tools and other different API (mail, translate, books etc.). API – application programming interface, set of built-in functions providing for outside usage. All of this performs on google servers. Cloud Machine Learning Engine brings the
power and flexibility of TensorFlow, scikit-learn and XGBoost to the cloud [9]. Google Cloud Natural Language API as a part of machine learning section enables to operate with text and voice. In particular text processing of this service include sentiment, entity, syntax and categories analysis. In this work will be submitted sentiment analysis of russian language news using Google NLP API [10]. Payment card is required on Google account for getting access to their platform. After card registration Google gives free 300 dollar for 1 year. Cloud Natural Language only support Chinese, English, French, German, Italian, Japanese, Korean, Portuguese and Spanish languages. Firstly, news will be translated into English through Translate API for deal with Russian text. Then translated text will be given to Cloud NLP as an input for sentiment analysis. Since this is a web API, all function calls are a separate http-request to the server. For security purposes need to get the already generated API key and attach it as a parameter to the request. The server will not be execute the request without it. The server sends structured data object as a response. This object consist of two parameter: there are score and magnitude. The score can be any value from -1 to 1, which is respectively from negative to positive, and mean values are neutrality. The second value is the magnitude, it shows the strength of emotions in the sentence, and can be from 0 to infinity. The greater the magnitude, the more emotionality of the text [11]. The cost of this service is up to $ 1 per million characters.

Figure 5. Data flow chart.

It is necessary to make 2 queries in total for each text. The first query for translation of the text from Russian to English, and the second for determination of tonality of the text. (Figure 5). Cloud NLP already has a trained model for sentiment analyzing of the text. Therefore only test data were submitted to Translate API. 5000 texts of 80 words each due to the limitations of the API for the number of symbols for translation. The histogram of the distribution of the number of words in the text indicates that the majority of news has at least 200 words (Figure 6).

Table 3. Classification Report for Google NLP with neutral, positive and negative.

| Class   | Precision | Recall | F-1 score |
|---------|-----------|--------|-----------|
| Negative| 0.51      | 0.41   | 0.45      |
| Positive| 0.32      | 0.56   | 0.41      |
| Neutral | 0.50      | 0.43   | 0.46      |
| Average/Total | 0.47     | 0.44   | 0.45      |

The test was carried out without pre-processing of texts. Red line shows a line of 80 words, yellow line – 200 words. There were two experiments. First, all the types of news was submitted – positive, negative and neutral. The result of entering the table 3. Accuracy is 44.44%. The next experiment was already without neutral news. Accuracy increased to 75.05%. Table 4 from the quotations.
Table 4. Classification Report for Google NLP only positive and negative.

| Class       | Precision | Recall | F-1 score |
|-------------|-----------|--------|-----------|
| Negative    | 0.80      | 0.56   | 0.60      |
| Positive    | 0.63      | 0.56   | 0.60      |
| Average/Total | 0.74      | 0.75   | 0.75      |

Figure 6. Distribution by number of words in texts.

As described on the tables above Google Cloud NLP show more accurate result excluding neutral texts. As on the plot below (Figure 7) the distribution of sentiment results by Google Cloud NLP show that superior majority of texts are neutral with value 0. Probably, 80 words per text not enough to determine tonality of the big texts.

6. Conclusion
In conclusion, it can be noted that Traditional methods of text classification works average on Russian texts and using TF-IDF works fairly well on sentiment analysis. Even though it requires a lot of preprocessing and model building it do not have limitations that have in most Google Services. Traditional algorithms like SVM and Naive Bayes works relatively well on text classification if TF-IDF is used as an embedding. As reported above these 2 algorithms are opposite of each other. SVM with high precision on negative and positive classes whereas, Naive Bayes has high precision on neutral class.

Acknowledgments
The work was funded by grant No. BR05236839 of the Ministry of Education and Science of the Republic of Kazakhstan.
Figure 7. Distribution of sentiment analysis score.

References
[1] Spärck Jones K 1972 A Statistical Interpretation of Term Specificity and Its Application in Retrieval Journal of Documentation 28 11–21
[2] Morphological analyzer (POS tagger + inflection engine) for Russian and Ukrainian languages
[3] Named Entity Recognition for Russian Language
[4] Rish I 2001 An empirical study of the naïve Bayes classifier IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence
[5] Cristianini N and Shawe-Taylor J 2000 An Introduction to Support Vector Machines and Other Kernel-based Learning Methods (Cambridge University Press)
[6] Cohen J, Cohen P, West S G and Aiken L S 2003 Applied multiple regression/correlation analysis for the behavioral sciences (2nd ed Hillsdale, NJ: Lawrence Erlbaum Associates)
[7] Quinlan J R 1987 Simplifying decision trees International Journal of Man-Machine Studies 27 (3) 221
[8] Hastie T, Tibshirani R and Friedman J 2009 Chapter 15 Random Forests The Elements of Statistical Learning: Data Mining, Inference, and Prediction
[9] Google Cloud Machine Learning Engine
[10] Kaggle competition, dataset of Russian news by Alem research group
[11] Google Cloud Natural Language