Applying of Sentiment Analysis for texts in Russian based on Machine Learning approach

M. V. Boyko
Department of Informatics and Robotics
Ufa State Aviation Technical University
Ufa, Russia
e-mail: russian_max@inbox.ru

D.R. Bogdanova
Department of Informatics and Robotics
Ufa State Aviation Technical University
Ufa, Russia
e-mail: dianochka7bog@mail.ru

A. Hilbert
Fakultät Wirtschaftswissenschaften
Technische universität Dresden
Dresden, Germany
e-mail: hilbert@wiid.wiwi.tu-dresden.de

Abstract[1]

This paper considers the problem of Sentiment classification in text messages in Russian with using Machine Learning methods - Naive Bayes classifier and the Support Vector Machine. *Text analysis, analysis of tonality; sentiment analysis, machine learning.*

1. Introduction

The present stage of human development is characterized by rapid growth of information. One of the most common forms of storage is the text in natural language. Textual form of information is natural for human beings and they readily accept it. The development of information technologies is accompanied by intense growth in the number of websites, which currently stands at more than 285 millions, and as a consequence of increasing the volume of text data. The vast amount of information collected in numerous text databases that are stored in personal computers, local and wide area networks. Average user is becoming more difficult to work with huge amounts of data. Reading the texts of the volume, manual search and analysis of relevant information in giant arrays of text data are ineffective. To solve this problem, and process automation was developed area of natural language processing, information retrieval, machine translation, information extraction , sentiment analysis and others. The article is devoted to the Sentiment Analysis of Russian text messages using Machine Learning. Sentiment Analysis in the text is one of the directions in the analysis of natural language texts. Sentiment is the emotional score, which is expressed in the text. It can have one-dimensional emotive space (two classes of sentiments) or multivariate (more than two). Foresight sentiment of the text lies in the fact that, based on textual information, it allows you to evaluate the success of the campaign, political and economic reforms, to identify relevant press and media to a certain person, to an organization for the event, to determine how consumers relate to a particular product, to services to the organization. In [1] the authors consider applying Sentiment Analysis to the study opinions of consumer of different banks.Despite the promise of this direction, while it is not as actively used in text processing systems. The reasons are the difficulties of highlight the emotional vocabulary in the texts, a imperfection of the existing text analyzers, dependence on the domain. Therefore, the improvement and development of new analytical methods based on machine learning is an urgent task.

The article presents the results of a study of Sentiment classification of texts in Russian with using Machine Learning.

2. Approaches to the Sentiment Analysis

There are three approaches of Sentiment Analysis of text messages.

1) Sentiment Analysis based on pre-defined dictionaries of tonality with linguistic analysis. Tonality dictionaries consist of elements such as words, phrases, patterns, each of which has its own emotional coloring. Tonality of the text is determined by the combination of emotive language found and evaluated in text. 2) Sentiment Analysis based on methods of Machine Learning. The text presents in vector form. According to the available

[1] **Proceedings of the 14th international workshop on computer science and information technologies CSIT'2012, Ufa – Hamburg – Norwegian Fjords, 2012**

training data is trained classifier. After that, it possible to classify sentiment in new text message.

3) The combination of the first and the second approaches.

The first approach is rather time-consuming because of the need for a tonality of dictionaries, a list of tonality patterns and the development of language parsers, but it is more flexible. The advantage of this approach is that it allows you to see the emotional vocabulary at the level of the sentence. In [2], the authors present an algorithm for Sentiment Analysis based on the tonal dictionaries consisting of several steps: morphological analysis of text mark-up vocabulary lists for the tonality vocabulary, syntactic analysis, and directly determine the tonality. The algorithm can be estimated on the website [3]. In [4], the authors developed the following algorithm for estimating the tonality of the text, which includes recognition of the object of tonality, parsing text, selection and classification of propositions that express the tonality, the assessment based on the general tonality of all the tonality propositions. Abroad, as well being an active search for and improve the analysis of tonality on the basis of tonality dictionaries and linguistic analysis. One such study is presented in [5]. It describes the analyzer, which consists of: 1) remove the special terminology of the text, and 2) determine the tonality, and 3) analysis of the associative relationship. The analyzer uses two linguistic system: a dial tonality dictionary and database templates. The approach is based on using Machine Learning, presupposes the existence of pre-marked-up the training set of data. The purpose of training in Sentiment Analysis is to get the necessary and sufficient rules, which you can use to make a classification of tonality of the new text messages, similar to those that made up the training set. The drawback of algorithms based on Machine Learning is dependence on the quality and quantity of training data. This approach does not allow an in-depth analysis of the text, to identify the object and the subject of tonality. Machine Learning methods for solving the problem of Sentiment classification of messages are actively developing overseas. In the Russian practice of science are not yet known cases of successful application of Machine Learning to Sentiment Analysis. Therefore, we consider some of the work of foreign authors.A great contribution to the development of Sentiment Analysis of text messages contributed by researchers from Cornell University B. Pang and L. Lee. In 2008 they published the book «Opinion Mining and Sentiment Analysis» [6] devoted to modern methods and approaches to Sentiment Analysis in text messages. In their paper [7] a Sentiment classification using Machine Learning and show that this approach is superior to a simple technique based on the compilation of dictionaries of commonly used positive and negative words. In their further work [8], the authors describe an algorithm that allows us to classify sentiments using only subjective sentences. Objective proposals generally do not have the emotional coloration, but create noise in the data.

In [9], the authors consider the problem that from the training data extracts a very large number of terms. The authors describe methods for selecting the most informative terms, and evaluation of their tonality. To address the shortcomings of the above approaches is used to combine them. Thus, in [10] method is based on the extracted lexical rules, while training with the participation of man and machine learning are combined into a sentiment classification algorithm. In another study [11], researchers from Microsoft suggest ways to get sentiment patterns using proposed algorithm. The result is achieved through automatic extraction of informative patterns with subsequent evaluation of tonality, combining with SVM. The combined approach is promising as it combines advantages of the first two approaches. Here, an important task for the study is to determine how they interact.

3. Description of algorithms for Sentiment Analysis

In this work we considered algorithms which based on using Machine Learning approach. As Machine Learning algorithms we chose a Naive Bayesian classifier and Support Vector Machine. For improving the accuracy of classification we considered a Meta-Machine Learning algorithm - Bagging for Naive Bayesian classifier. Mathematically, the problem of classifying of sentiment can be represented as follows. There are two classes - the class of positive messages c_1 and class of negative messages c_2 (1): $C = \{c_1, c_2\}$, (1)

there are set of messages (2):

$$D = \{d_1, d_2, ..., d_n\}, \quad (2)$$

and an unknown classification function (3):

$$F : C \times D \to \{0,1\}, \quad (3)$$

we need to build a classifier F' as close to the classification function F as possible. We have a labelled set of messages for learning (4).

$$K \subset C \times D^l \quad (4)$$

Feature space in the this problem can be represented using the vector model. Each text message is treated as a set of words ("bag" of words). This view of a text message is presenting as a point in multidimensional space. Points lying close to each other correspond to semantically similar messages. In this model, a sequence of words is ignored. For example, the "*хорошая книга*" and "*книга хорошая*" is the same. Thus, the message is a "bag" with the words.

3.1 Naive Bayes classifier

Let's consider Naive Bayesian classifier for sentiment classification problem. Let each message d takes the

values from the dictionary V, and is described by a set of words $\{w_1, w_2, \ldots, w_n\}$. There is a set of classes $C = \{c_1, c_2\}$, consisting of a class of positive messages and a class of negative messages. We need to find the most probable value of the corresponding class of the set of words (5):

$$c_{NB} = \arg\max_{c_j \in C} p(d = c_j \mid w_1, w_2, \ldots, w_n) \quad (5)$$

It is known that the conditional probability of an event can be found using the Bayes theorem (6):

$$p(d = c_j \mid w_1, w_2, \ldots, w_n) = \frac{p(w_1, w_2, \ldots, w_n \mid d = c_j) \cdot p(d = c_j)}{p(w_1, w_2, \ldots, w_n)} \quad (6)$$

Then the expression (5) takes the form (7):

$$c_{NB} = \arg\max_{c_j \in C} \frac{p(w_1, w_2, \ldots, w_n \mid d = c_j) \cdot p(d = c_j)}{p(w_1, w_2, \ldots, w_n)} \quad (7)$$

From the expression (7), we are interested only in the numerator, because the denominator does not depend from the class. Thus, the denominator is a constant and can be reduced. Assuming conditional independence of attributes, we obtain the expression (8) which is using for classification:

$$c_{NB} = \arg\max_{c \in C} p(w_1, w_2, \ldots, w_n \mid d = c) \cdot p(d = c) \quad (8)$$

Naive Bayesian classifier operates under the following assumptions:

- Words and phrases in the message are independent from each other;
- Do not takes into account the sequence of words;
- Do not takes into account the length of the message.

There are two ways to implement a Naive Bayesian classifier – a Bernoulli model and multinomial model. Difference is that in the Bernoulli model is considering a presents of word in message. In the multinomial model is considering the number of occurrences of a word in the text. Table 1 provides an example of a vector notation of the text.

Table 1

Example of vector form

	Vector description
Bernoulli model	[0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0]
Multinomial model	[0, 0, 2, 1, 0, 3, 1, 2, 0, 0, 0]

3.1.1 Bernoulli model

Let's consider the sentiment classification algorithm with the Bernoulli model [12]. In the Bernoulli model, the message described by the vector consisting of the attributes with values 0 or 1. Thus, we consider only the presence or absence of words in the message and how many times it is repeated in the message is not important.

Given a vocabulary $V = \{w_t\}_{t=1}^{|V|}$. Then the message d_i describes by the vector of length $|V|$, consisting of bits b_{it}. If a word w_t appears in the message d_i then $b_{it} = 1$,, if not then $b_{it} = 0$. Then the likelihood of belonging to a class c_j of messages d_i can be calculated by the formula (9):

$$p(d_i \mid c_j) = \prod_{t=1}^{|V|} (b_{it} \cdot p(w_t \mid c_j) + (1 - b_{it}) \cdot (1 - p(w_t \mid c_j))) \quad (9)$$

For learning a classifier it needs to find the probabilities $p(w_t|c_j)$. Let there be a training set of messages $D = \{d\}_{i=1}^{|D|}$ which has labels of classes c_j, then it is possible to calculate estimates of the probabilities that a particular word occurs in a particular class (10):

$$p(w_t \mid c_j) = \frac{1 + \sum_{i=1}^{|D|} b_{it} \cdot p(c_j \mid d_i)}{2 + \sum_{i=1}^{|D|} p(c_j \mid d_i)} \quad (10)$$

A priori probabilities of classes can be calculated by the formula (11):

$$p(c_j) = \frac{\sum_{i=1}^{|D|} p(c_j/d_i)}{|D|} \quad (11)$$

Then, the classification will be carried out by the formula (12).

$$c_{NB} = \arg\max_{c_j \in C} p(c_j) \cdot p(d_i \mid c_j) = \arg\max_{c_j \in C} [\log(\sum_{i=1}^{|D|} p(c_j \mid d_i)) + \sum_{t=1}^{|V|} \log[b_{it} \cdot p(w_t \mid c_j) + (1 - b_{it})(1 - p(w_t \mid c_j))]] \quad (12)$$

From (10) follows, that the some probabilities will be zero, since that some words can be presented in one class of training data and can be absent in another. Difficulties arise with zero probabilities when they are multiplied in (12). In this case the entire expression is zero and there is a loss of information. To avoid zero probability of obtaining used add-one, or Laplace smoothing, which consists of adding one to the numerator (13).

$$p(w_t \mid c_j) = \frac{1 + \sum_{i=1}^{|D|} b_{it} \cdot p(c_j \mid d_i)}{2 + \sum_{i=1}^{|D|} p(c_j \mid d_i)} \quad (13)$$

Sentiment classification algorithm using the Bernoulli model is shown in Figures 1 and 2. It consists of learning part and classifying part. In the learning part there are input parameters is a set of labelled messages and set of classes. In this part creates a dictionary of words V, that estimates $p(c_j)$ and $p(w_t|c_j)$, sets the threshold value h which minimize the classification error. Output is a fully trained classifier with set parameters. Classifying part applies for new message, which sentiment must be determined.

Input: set of documents D = $\{d_1, d_2, ..., d_n\}$,
set of classes C = $\{c_1, c_2\}$

1. Extract all terms from D to the vocabulary V
2. For each $c_j \in C$ do
   3. Count documents N^c in each class
   4. Calculate probability $p(c_j) = N^c/N$
   5. For each $w_t \in V$ do
      6. Count documents $N^c_{w_t}$ in class containing word w_t
      7. Calculate probability $p(w_t|c_j) = (N^c_{w_t} + +1)/(N^c + 2)$
8. Set threshold h with minimal classification error

Output: V, $p(c_j)$, $p(w_t|c_j)$, h

Fig.1. Algorithm of learning NB Bernoulli model

Input: document d, V, $p(c_j)$, $p(w_t|c_j)$, h

1. Extract all terms from d to the vocabulary V_d
2. For each $c_j \in C$ do
   3. score$[c_j] = \ln p(c_j)$
   4. For each $w_t \in V$ do
      5. If $w_t \in V_d$ than score$[c_j]$ += $\ln p(w_t|c_j)$
      6. else score$[c_j]$ += $\ln(1 - p(w_t|c_j)$
7. If score$[c_1] > h \cdot$ score$[c_2]$ than $d \in c_1$ else $d \in c_2$

Output: tonality of document d

Fig. 2. Algorithm of classification
NB Bernoulli model

3.1.2 Multinomial model

In the multinomial model [12], the message is a sequence of random selection of some word from the dictionary. This model takes into account the number of repetitions of each word in a one message, but ignores words that are absence in the message. Given a vocabulary $V = \{w_t\}_{t=1}^{|V|}$. Then message d_i can be described by the vector of length $|V|$, consisting of words, which is taken from the dictionary with probability $p(w_t|c_j)$. Then the likelihood of belonging of messages d_i to a class c_i estimates by formula (14).

$$p(d_i \mid c_j) = p(|d_i|) \cdot |d_i|! \cdot \prod_{t=1}^{|V|} \frac{1}{K_{it}!} p(w_t \mid c_j)^{K_{it}}, \quad (14)$$

where K_{it} - is the number of occurrences of word w_t in the message d_i.

For learning the classifier it also needs to find the probabilities $p(w_t|c_j)$. Let there be a training set of messages $D = \{d\}_{i=1}^{|D|}$, which is distributed in classes c_i and we know the number of occurrences of words in the message K_{it}. Then we can calculate estimates of the probabilities that a particular word occurs in a particular class (15). In this case, also apply smoothing add-one.

$$p(w_t \mid c_j) = \frac{1 + \sum_{i=1}^{|D|} K_{it} \cdot p(c_j \mid d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} K_{is} \cdot p(c_j \mid d_i)} \quad (15)$$

A priori probabilities of classes can be calculated by the formula (16).

$$p(c_j) = \frac{\sum_{i=1}^{|D|} p(c_j \mid d_i)}{|D|} \quad (16)$$

Then, the classification will be carried out by the formula (17).

$$c_{NB} = \arg\max_j p(c_j) \cdot p(d_i \mid c_j) = \\ = \arg\max_j [\log(\sum_{i=1}^{|D|} p(c_j \mid d_i)) + \\ + \sum_{t=1}^{|V|} K_{it} \cdot \log p(w_t \mid c_j)] \quad (17)$$

Classification algorithm with the Multinomial Naïve Bayes model is shown in Figures 3 and 4. It consists of learning part and Sentiment classification part. In the learning part creates a dictionary of terms V, estimates probabilities $p(c_j)$ and $p(w_t|c_j)$, set the threshold value of h, to minimize the classification error. Classifying part applies for new message, which sentiment must be determined.

Input: set of documents D = $\{d_1, d_2, ..., d_n\}$,
set of classes C = $\{c_1, c_2\}$

1. Extract all terms from D to the vocabulary V
2. For each $c_j \in C$ do
   3. Count documents N^c in each class
   4. Calculate probability $p(c_j) = N^c/N$
   5. For each $w_t \in V$ do
      6. Count number of occurrences $M^c_{w_t}$ of word w_t in each class
      7. Calculate probability $p(w_t|c_j) = (M^c_{w_t} + +1)/\sum_{t=1}^{|V|}(M^c_{w_t} + 1)$
8. Set threshold h with minimal classification error

Output: V, $p(c_j)$, $p(w_t|c_j)$, h

Fig. 3. Algorithm of learning NB Multinomial model

Input: document d, V, $p(c_j)$, $p(w_t|c_j)$, h

1. Extract all terms from d to the vocabulary V_d
2. For each $c_j \in C$ do
   3. score$[c_j] = \ln p(c_j)$
   4. For each $w_t \in V$ do
      5. If $w_t \in V_d$ than score$[c_j]$ += $\ln p(w_t|c_j)$

6. If $score[c_1] > h \cdot score[c_2]$ than $d \in c_1$
else $d \in c_2$

Output: tonality of document d

Fig. 3. Algorithm of learning NB Multinomial model

3.1.3 Bagging algorithm

One of the algorithms for improving the quality of classification is called Bagging. It was proposed by L. Breiman and describes in [16]. Bagging algorithm is shown in Figure 7.

Input: $(d_1, y_1), (d_2, y_2), \ldots, (d_m, y_m)$, where $d_i \in D$ – set of documents;
$y_i \in Y = \{+1, -1\}$, $y_i = +1$ if positive and $y_i = -1$ negative polarity of documents;
T – number of classifiers in ensemble;
$|D|$ - length of training data;
e – admissible error of classification

1. For $t = 1$ *to* T do
   2. Choose randomly $|D|$ documents for constructing training dataset D_t from D
   3. Construct control dataset from D/D_t
   4. Construct classifier $h_t: D \rightarrow \{+1, -1\}$
   5. estimate error e_t of classifier h_t on dataset D/D_t
   6. If $e \geq e_t$ then add classifier h_t into ensemble

Output: $H(d) = sign(\sum_{t=1}^{T} h_t(d))$ final ensemble of classifiers

Fig. 7. Bagging algorithm

From the initial training set of D of length $|D|$ forms training subsets D_t of the same length $|D|$ with the bootstrap - a random selection with returns. However, some messages will appear in a subset of a few times, some - not even once. Next, set the control messages by subtracting D/D_t. With using training subset D_t learns classifier h_t. Classification error e_t of h_t estimates by the control subset D/D_t and then compared with the admissible error of the classification of e. If the error is less than a classifier built admissible error, then it is added to the ensemble. Sentiment classification producing with ensemble of classifiers by a simple voting.

3.2 Support Vector Machine

The main idea of Support Vector Machine algorithm is to find separating hyperplane, represented by vector $\overline{w}$ which minimize empirical error of classification and maximize margin between classes. SVM was proposed by V. Vapnik and A. Chervonenkis. SVM is a high effective in classification problems and has popularity among Machine Learning algorithms. In particularly, it outperforms other algorithms of Machine Learning in text categorization. The finding of separating hyperplan corresponds to a constrained quadratic optimization problem. Let $c_j \in \{1, -1\}$ be the class of document d_j, then the solution can be written as (18):

$$\overline{w} = \sum_j \alpha_j c_j \overline{d}_j, \quad \alpha_j \geq 0 \qquad (17)$$

Where α_j are obtained by solving a dual optimization problem. Those $\overline{d}_j$ such that α_j is greater than zero are called support vectors, since they are the only document vectors contributing to $\overline{w}$. Classification of message consists of determining which side of $\overline{w}$ hyperplan it fall on.

The main disadvantage of Support Vector Machine that it has cubic complexity in the size of dataset and requires a lot of computational recourses. The cause is that it have to solve quadratic optimization problem with the number of parameters equal to number of data and to compute dot product many times.

There are many modifications of SVM developed for reducing computational time. One of them is Sequentional Minimal Optimization algorithm [17] developed by J. Platt. This algorithm is used in this work. It allowed to receive the results in acceptable time.

In this work realized to variants of SVM – first variant considers only a presents/absence of features and in the second variant considers the number of occurrences of features.

4. Results of experiments

In this research we aimed to study a few points of Sentiment classification:

- Evaluate performance for Sentiment classification of text messages in Russian language;
- Compare performance with results obtained for text messages in English language;
- Study influence of lemmatization on the accuracy of classification;
- Study influence of a length of word on the accuracy of classification;
- Study influence of the grouping words which have equal semantic meaning on the accuracy of classification.

According these aims was developed a program «Text Analyzer» in the program language C#. All listed algorithms of Sentiment classification were realized in this program.

For learning and evaluation of the accuracy of the sentiment classification was used the test set, consisting of customer reviews of a few Russian banks taken from the Internet site [13]. It includes 304 positive reviews and 850 negative reviews in Russian. An example of review with a positive sentiment is: "An application for a loan designed to quickly, no questions asked, within 20 minutes." An example of negative review: "Consideration of the application took time for two months". For evaluation of Sentiment classification for text in English was used dataset includes 1000 negative and 1000 positive reviews about films from IMDB [18].

For study influence of lemmatization in the pre-processing text was entered lemmatization of all

occurring words. Lemmatization brings different words to their initial form, for example, the noun is the nominative case, singular. Motivation for lemmatization of the text due to the fact that different forms of word can often express the same meaning. In this regard, is justified to bring the words to a initial form. We used LemmaGen library written in C# and designed for lemmatization of words. These libraries are available on the website of the developer [14].

To evaluate the generalization capability of the algorithm used by a sliding control or cross-validation. Fixed set consisting of 10 partitions of the original sample, each of which in turn consisted of two subsamples: the training and control. For each partition, configures the algorithm for the training subsample, and then evaluated its average error on the objects of the control subsample. Assessment of the sliding control was averaged over all partitions of the error on the control subsamples. For the bagging algorithm accepted allowable error of the classifier is equal to e = 25%.

To evaluate the classification accuracy of each control unit is used indicator "classification accuracy", which is calculated by the formula (18):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \cdot 100\%, \quad (18)$$

where TP - the number of correctly classified positive messages; TN - the number of correctly classified negative messages; FP - the number of non correct classified positive messages; FN - the number of non correct classified negative messages.The results of computational experiments presented in the Table 2. Accuracy of classification lies in range 85% - 88,3%. For Naïve Bayes the best results obtained by Multinomial model 86,83% (Bernoulli – 86,49%) or another words by considering number of occurrences of words. Using Bagging algorithm has a positive influence on the classification. It improved accuracy for NB Multinomial model by 0,86%. For Support Vector Machine with leaner core the best results obtained by considering presence of word in the message (87,69% vs. 85%). Using of polinomial core gave 86,73% of accuracy. For Sentiment classification we received better results on dataset in Russian on an average 5%. This suggests that dataset in Russian has more constrained domain - banking. In contrast, dataset in English has widely range of different words, because most reviews have a description of film story. Analysis of results also indicates that SVM outperforms Naïve Bayes algorithm in two cases of language.

Table 2

Results of experiments with texts in Russian

Naïve Bayes classifier	
NB Bernoulli model	86,49%
NB Multinomial model	86,83%
NB Multinomial model with synonyms	86,93%
NB Multinomial model, length > 2	86,40%
Bagging NB Bernoulli model (e=25%)	86,82%
Bagging NB Multinomial model (e=25%)	87,69%
Support Vector Machine	
SVM, presence, leaner	87,69%
SVM, occurance, leaner	85,00%
SVM, presence, leaner, without lemmatizator	87,07%
SVM, presence, leaner, length > 2	88,21%
SVM, presence, leaner, with synonyms	87,77%
SVM, presence, polinomial	86,73%
SVM, presence, leaner, with synonyms, length > 2	88,30%

Table 3

Results of experiments with texts in English

Naïve Bayes classifier	
NB Bernoulli model	80,25%
NB Multinomial model	81,05%
Support Vector Machine	
SVM, presence, leaner	84,3%
SVM, occurance, leaner	83,15%
SVM, presence, leaner, without lemmatizator	85,85%

For grouping words which have the same semantic meaning was used vocabulary of synonyms with 5371 strings. For example, if in message occur to different words "borrow" and "lend" then it is equivalent occurring two words "borrow". This modification allowed to improve accuracy by 0,1% for NB, and 0,08% for SVM. It is not so much but we hope that using more bigger and specific vocabulary of synonyms can give a more significant effect. Lemmatization has positive influence for Sentiment classification of text in Russian (87,07% without lemmatization, 87,69% with lemmatization). It could be explained that in Russian language words could have different endings. Lemmatization allows to group cognate words with one semantic meaning and different endings. In text in English the best result received without lemmatization (84,3% vs. 85,85%).

For excluding prepositions and articles from feature words was made experiment in which considering words with length more than two letters. This modification gave improving on 0,52% in SVM. But in NB accuracy descended on 0,43%. The best result 88,30% received by SVM with lemmatization, grouping synonyms and length of word >2.

5. Conclusion

Based on results of Sentiment classification of texts in Russian we obtained the following conclusions;

- Machine Learning could provide accuracy of sentiment classification 85% - 88,3% for considered texts in Russian;

- SVM confirmed that it outperforms Naïve Bayes algorithm in two cases of language;
- Multinomial model surpasses Bernoulli model in NB;
- Bagging algorithm has a positive influence on the classification but little;
- presence feature of words surpasses number of occurance in SVM;
- using synonyms has positive influence on Sentiment classification but little;
- lemmatization has positive influence for Sentiment classification of test in Russian, but not for text in English.

The task of sentiment classification of text messages has a complex nature and requires innovative approaches for solution. The complexity of its nature is that the initial data are the texts in natural language. Every word of this text has its meaning, and the combination of words is a complex interaction of the meaning of each word. At present there is no universal method of modeling such an interaction in the language of the machine or the language of numbers. Despite the complexity of the problem, it attracts a large number of researchers around the world. Searches in this area are actively maintained and there are some achievements. Many of the developed algorithms achieve classification accuracy greater than 85%. But keep in mind that these results were obtained on test data under experimental conditions. Unfortunately there is no official information about the real successful practical application of systems to solve such problems.

Acknowledgements

The research is supported by the Russian Fond of Foundation Research grant № 12-07-00377-a, the research work is performed within the state work on development of software tools support decision-making for different kind of management activity in industry in the conditions semi structured data based on the technology of distributed artificial intelligence.

References

1. Boyko M., Hilbert A., Yussupova N., Bogdanova D. Marketing research of cosumer opinions with using information technologies. Proceedings of the 13th International Workshop on Computer Science and Information Technologies, Germany, Garmisch-Paterkirchen, September 27 - October 02, 2011, p 103-105.
2. Pazelskaya A.G., Soloviev A.N. Method of the determination emotions in the lyrics in Russian / / computer program linguistics and intellectual technologies. Sat scientific articles / Issue 10 (17). - Moscow: Publishing House of the State Humanitarian University, 2011. - S.510-522.
3. The official site of "Эр Си О". Component of the definition of key texts [electronic re-LAS] (http://x-file.su/tm/Default.aspx)
4. Ermakov A.E., Kiselev S.L. Linguistic model for the computer analysis of key media of publications. Computational Linguistics and the intellectual technology: proceedings of the International Conference Dialog'2005. - Moscow: Nauka, 2005.
5. Yi J. Nasukawa T., Niblack, W., Bunescu, R. Sen-timent analyzer: Extracting sentiments about a given topic using natural language processing techniques. In Proceedings of the 3rd IEEE international conference on data mining (ICDM 2003) Florida, USA, (pp. 427-434).
6. Pang B., Lee L. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval. Vol. 2, Nos. 1-2 (2008), 1-135.
7. Pang B., Lee L. Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of the Conference on Empirical Methods in Natural. Language Processing (EMNLP), Philadelphia, July 2002, pp. 79-86.
8. Eight. Pang B., Lee LA Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Mini-mum Cuts. Proceedings of the ACL, 2004.
9. O'Keefe T., Koprinska I. Feature selection and weighting methods in sentiment analysis. Australasian Doc-ument Computing Symposium, 2009.
10. Prabowo R., Thelwall M. Sentiment analysis: A combined approach. Journal of Informatics in 2009.
11. König A., Brill E. Reducing the Human Overhead in Text Categorization. Proceedings of KDD, 2006.
12. Manning C., Raghavan P., Schuetze H.. An Intro-duction to Information Retrieval. Cambridge University Press. Cammbridge, England (2009).
13. Internet portal dedicated to the Russian ban-kam [electronic resource] (http://banki.ru).
14. Portal dedicated to stemming [Electron-HYDRATED resource] (http://lemmatise.ijs.si/Software/Version3).
15. Freund Y., Schapire R. Experiments with New Boosting Algorithm. Machine Learning: Proceedings of the Thirteenth International Conference, pages 148-156, 1996.
16. Breinman L. Bagging Predictors. Machine Learning, 24, 123-140 (1996).
17. J. Platt. "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines". *Microsoft research, Technical Report MSR-TR-98-14, April 21, 1998.*
18. http://www.cs.cornell.edu/People/pabo/movie-review-data/