SVM and Naïve Bayes Classification Ensemble Method for Sentiment Analysis

Konstantinas KOROVKINAS[1], Paulius DANĖNAS[2], Gintautas GARŠVA[1]

[1] Institute of Applied Informatics, Kaunas Faculty, Vilnius University, Muitines Str. 8, Kaunas, Lithuania
[2] Center of Information Systems Design Technologies, Department of Information Systems, Kaunas University of Technology, Studentu Str. 50-313a, Kaunas, Lithuania

konstantinas.korovkinas@knf.vu.lt, paulius.danenas@ktu.lt, gintautas4garsva@gmail.com

Abstract. This paper follows an important problem of sentiment recognition which may influence ones decisions or reviews about item and etc. In this paper we introduce a new method to improve classification performance in sentiment analysis, by combining SVM and Naïve Bayes classification results to recognize positive or negative sentiment, and test in on datasets from movie reviews, sentiment140 and Amazon reviews. This method is evaluated on a training dataset which consists positive and negative words, and hold-out testing dataset, as well with training data from the same area. It was observed that better results were obtained using our proposed method in all the experiments, compared to simple SVM and Naïve Bayes classification.

Keywords: sentiment analysis, SVM, Naïve Bayes classification

1 Introduction

The concept of sentiment analysis and opinion mining were first introduced in the 2003. Sentiment analysis became very popular since people started using Facebook, Twitter, Instagram and other social networks. The main goal of research related to sentiment analysis is to obtain authors feelings expressed in positive or negative comments. This analysis is performed in multiple levels (Thomas, 2013): document level (Yessenalina et al., 2010), sentence level (Farra et al., 2010), word/term level (Engonopoulos et al., 2011) or aspect level (Zhou and Song, 2015; Hussein, 2016). This topic is considered as very challenging - although a lot of work has been done in this field, accuracy is still rather average due to comments, slang, smiles and etc.

Traditionally, sentiment classification can be regarded as a binary-classification task (Pang et al., 2002; Dave et al., 2003). Dave et al. (2003) use structured reviews for

testing and training, identifying appropriate features and scoring methods from information retrieval for determining whether reviews are positive or negative. These results perform as well as traditional machine learning method then use the classifier to identify and classify review sentences from the web, where classification is more difficult (Khainar and Kinikar, 2013).

Liu (2010), Tang et al. (2009) expressed an overview in sentiment analysis in which analyzed the strong points and the weak points of sentiment analysis and they gave many research ways of sentiment analysis. Barbosa and Feng (2010) pointed that n-gram is slow, so they researched on Microblogging features (Le and Nguyen, 2015).

Pang et al. (2002), Pang and Lee (2008) compared many classifiers on movie reviews and gave a vision of insight and comprehension in sentiment analysis and opinion mining. Authors also used star rating as a feature for classification (Le and Nguyen, 2015). In another paper Pang et al. (2002) evaluated the performance of Naïve Bayes, maximum entropy, and support vector machines in the specific domain of movie reviews, obtaining accuracy slightly above 80%. Go et al. (2009) later obtained similar results with unigrams by introducing a more novel approach to automatically classify the sentiment of Twitter messages as either positive or negative with respect to a query term. The same techniques were also used in Kharde and Sonawane (2016) to perform sentiment analysis on Twitter data, yet resulting in lower accuracy; again, SVM proved to perform best. Davidov et al. (2010) also stated that SVM and Naïve Bayes are best techniques to classify the data and can be regarded as the baseline learning methods, by applying them for analysis based on the Twitter user defined hashtag in tweets. The features were obtained after preprocessing step using the ngrams, punctuation, single words and pattern as different feature types and then combined in to a single feature vector for the classification. K-nearest neighbor strategy was used to assign labels in each training and testing data set.

Wan and Gao (2015) applied an ensemble sentiment classification strategy based on Majority Vote principle of multiple classification methods, including Naïve Bayes, SVM, Bayesian Network, C4.5 Decision Tree and Random Forest algorithms, for sentiment classification of twitter data for airline services. The results show that the proposed ensemble approach outperforms individual classifiers in this airline service Twitter dataset. Manek et al. (2017) propose a Gini Index based feature selection method with Support Vector Machine (SVM) classifier for large movie review data set sentiment classification. Results were compared with other feature selection methods on movie reviews and results have shown that classification by using this efficient and novel method has improved the accuracy.

Catal and Nangir (2017) proposed a novel sentiment classification technique based on Vote ensemble classifier utilizes from three individual classifiers: Bagging, Naïve Bayes, and Support Vector Machines (SVM), for Turkish sentiment classification problem. Proposed approach achieved better performance than Naïve Bayes, which was reported the best individual classifier for used datasets, and Support Vector Machines.

Such results led to the conclusion that SVM and Naïve Bayes, and combination of these methods are still prominent for future research and can be used to develop new hybrid techniques based on their combination. Therefore, in this paper we introduce a new method based on Naïve Bayes classification and SVM to recognize positive or

negative sentiments and compare accuracy with the baseline techniques. The rest of the paper is organized as follows. In section 2, a description of SVM and Naïve Bayes classification which were used in the experiment. In section 3, introduced our method. In section 4, described preparation of dataset, experimental settings and results. In section 5, we conclude and give tasks of our future works.

2 Relevant machine learning algorithms

2.1 Naïve Bayes classification

A Naïve Bayes classifier is a simple probabilistic classifier based on Bayes' theorem and is particularly suited when the dimensionality of the inputs are high. In text classification, the given document is assigned a class

$$C^* = arg \max_c \ p(c|d)$$

Its underlying probability model can be described as an "independent feature model". The Naïve Bayes (NB) classifier uses the Bayes' rule Eq. (1),

$$p(c|d) = \frac{p(c)p(d|c)}{p(d)} \tag{1}$$

Where, $p(d)$ plays no role in selecting C^*. To estimate the term $p(d|c)$, Naïve Bayes decomposes it by assuming the f_i's are conditionally independent given d's class as in Eq.(2),

$$p_{NB}(c|d) := \frac{p(c) \left(\prod_{i=1}^{m} p(f_i|c)^{n_i(d)} \right)}{p(d)} \tag{2}$$

Where, m is the no of features and f_i is the feature vector. Consider a training method consisting of a relative-frequency estimation $p(c)$ and $p(f_i|c)$ (Pang et al., 2002).

2.2 Support Vector Machines

Support vector machines were introduced in Boser et al. (1992) and basically attempt to find the best possible surface to separate positive and negative training samples. Support Vector Machines (SVMs) are supervised learning methods used for classification.

Given training vectors $x_i \in R^n$, $i = 1, \ldots, l$, in two classes, and an indicator vector $y \in R^l$ such that $y_i \in \{1,-1\}$, $C - SVC$ (Boser et al., 1992; Cortes and Vapnik, 1995) solves the following primal optimization problem (Chang et al., 2011).

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{3}$$

subject to $y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i,$

$$\xi_i \geq 0, \quad i = 1, \ldots, l$$

where $\phi(x_i)$ maps x_i into a higher-dimensional space and $C > 0$ is the regularization parameter. Due to the possible high dimensionality of the vector variable w, usually we solve the following dual problem.

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \tag{4}$$

subject to $\ y^T \alpha = 0,$

$$0 \leq \alpha_i \leq C, \quad i = 1, \ldots, l$$

where $e = [1, ..., l]^T$ is the vector of all ones, Q is an l by l positive semidefinite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, and $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is the kernel function.

After problem (4) is solved, using the primal-dual relationship, the optimal w satisfies.

$$w = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i) \tag{5}$$

and the decision function is

$$sgn(w^T \phi(x) + b) = sgn\left(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b\right)$$

(Chang et al., 2011)

3   The proposed technique

Our introduced methodology is focused on combine SVM and Naïve Bayes classification algorithms to get better results. In the figure below is presented system algorithm which show us principle of data processing from training data up to obtaining the results. Training and testing data had been preprocessed and cleaned before it was passed as the input of machine learning algorithms. It included removing redundant tokens such as hashtag symbols @, numbers, "http" for links, punctuation symbols, etc. Below are presented algorithms which are used in *"Combination"*. *"Results"* is the final results set with classified sentiments: "positive" or "negative".

Algorithm for words
Input: Let us denote the probability of word selection as p, and the threshold for its selection as th_1 (see subsect 4.2).
$D_{test} = \{S_1, S_2, \ldots, S_n\}$ - set of testing data
$S = \{w_1, w_2, \ldots, w_n\}$ - sentences
w - words which are contained in sentence
$R_{SVM} = \{SVMsent, v\}$ - set of SVM results, $SVMsent$ - sentiment
$R_{NB} = \{NBsent, v\}$ - set of Naïve Bayes classification results, $NBsent$ - sentiment
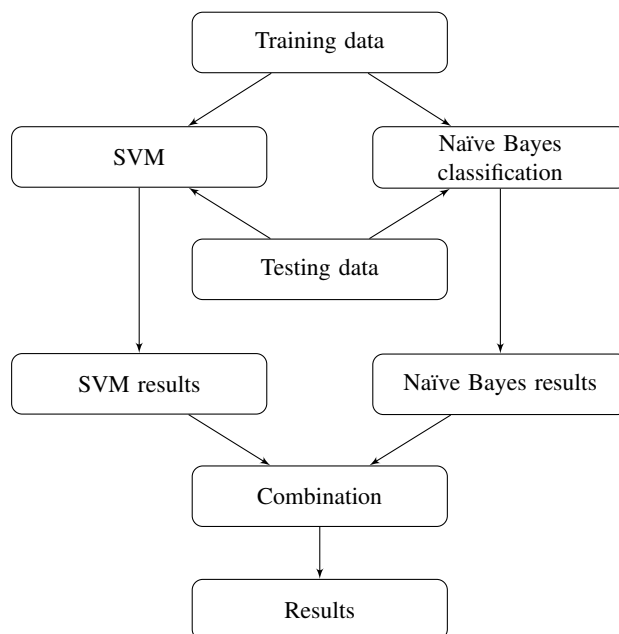v - value for Sum results

```
        ┌─────────────────┐
        │  Training data  │
        └─────────────────┘
       ↙                   ↘
┌───────────┐        ┌──────────────┐
│    SVM    │        │ Naïve Bayes  │
│           │        │classification│
└───────────┘        └──────────────┘
       ↘                   ↙
        ┌─────────────────┐
        │  Testing data   │
        └─────────────────┘
   ↓                           ↓
┌─────────────┐        ┌──────────────────┐
│ SVM results │        │ Naïve Bayes results│
└─────────────┘        └──────────────────┘
       ↘                   ↙
        ┌─────────────────┐
        │   Combination   │
        └─────────────────┘
                 ↓
        ┌─────────────────┐
        │     Results     │
        └─────────────────┘
```

Fig. 1. Proposed method for combining results.

1. SVM classification is performed:

$R_{SVM} = \{\}$

for $\forall S_i \in D_{test}$:

     $v_i = 0;$

     for $\forall w_j \in S_i$:

         pass w_j to SVM input (output $SVMsent_j$ and p_j)

         if $|p_j| \geq th_1$

            if $SVMsent_j <> "positive"$

               $p_j = -(p_j)$

     $v_i = v_i + p_j$

$$R_{SVM} = R_{SVM} \cup \{SVMsent_i, v_i\}, \quad SVMsent_i = \begin{cases} "positive", \text{if } v_i \geq 0 \\ "negative", \text{if } v_i < 0 \end{cases}$$

2. Naïve Bayes classification is performed:

$R_{NB} = \{\}$

for $\forall S_i \in D_{test}$:

     $v_i = 0;$

     for $\forall w_j \in S_i$:

         pass w_j to Naïve Bayes input (output v_j)

$$v_j = \begin{cases} 1, \text{for "positive" words} \\ -1, \text{for "negative" words} \end{cases}$$

     $v_i = v_i + v_j$

$$R_{NB} = R_{NB} \cup \{NBsent_i, v_i\}, \quad NBsent_i = \begin{cases} "positive", \text{if } v_i \geq 0 \\ "negative", \text{if } v_i < 0 \end{cases}$$

3. Results are combined as following:
   (a) Find results which are the same in both *SVM* and *Naïve Bayes classification*. $Results = R_{SVM} \cap R_{NB} = \{x : x \in R_{SVM}\{SVMsent\} \text{and } x \in R_{NB}\{NBsent\}\}$
   (b) Find results which are different between *SVM* and *Naïve Bayes classification*. $R_{SVM}\{SVMsent\}\Delta R_{NB}\{NBsent\}$
   (c) Find *coefficient of difference* for $\forall (R_{SVM}\{SVMsent\}\Delta R_{NB}\{NBsent\})$, using our proposed formula (we need to unify $R_{NB}\{v\}$ values, so we used $\log_{10}$ for it):

   $$difference = R_{SVM}\{p\} + \log_{10}\left(|R_{NB}\{v\}|\right)$$

   (d) Find *average* of all *coefficients of difference*.
   (e) for $\forall difference_i \in difference$:
   $$Results = \begin{cases} Results \cup R_{SVM}, \text{if } difference_i \leq average \\ Results \cup R_{NB}, \text{if } difference_i > average \end{cases}$$

Output: set of classification results $Results = \{S, sentiment\}$ and Accuracy (see subsect 4.3).

Algorithm for sentences This technique is applied on the whole sentence without splitting into words.
Input: Let us denote th_2 (see subsect 4.2) as the threshold value for $R_{SVM}\{p\}$ selection in algorithm step (b) and th_3 as the threshold to select $R_{SVM}\{p\}$ in algorithm step (c).
$R_{SVM} = \{SVMsent, p\}$ - set of SVM results obtained after performing *SVM* classification; $SVMsent$ - sentiment
p - the probability of sentence classification
$R_{NB} = \{NBsent, v\}$ - set of Naïve Bayes classification results obtained after performing *Naïve Bayes classification*; $NBsent$ - sentiment
v - Naïve Bayes results value, contains "1" for "positive" sentence and "-1" for "negative" sentence
$th_3 = \min(R_{SVM}\{p\}) + \sigma_{R_{SVM}\{p\}} - 0.01$ (used our proposed formula), where $\sigma_{R_{SVM}\{p\}}$ is the standard deviation of $R_{SVM}\{p\}$

Algorithm for results combining is performed:

1. Find results which are the same in both *SVM* and *Naïve Bayes classification*.
   $Results = R_{SVM} \cap R_{NB} = \{x : x \in R_{SVM}\{SVMsent\} \text{and } x \in R_{NB}\{NBsent\}\}$
2. Find results which are different between *SVM* and *Naïve Bayes classification*.
   $R_{SVM}\{SVMsent\}\Delta R_{NB}\{NBsent\}$ and $R_{SVM}\{p\} < th_2$
3. $Results = \begin{cases} Results \cup R_{SVM}, \text{if } |R_{SVM}\{p\}| < th_3 \\ Results \cup R_{NB}, \text{if } |R_{SVM}\{p\}| \geq th_3 \end{cases}$

Output: set of classification results $Results = \{S, sentiment\}$ and Accuracy (see subsect 4.3).

4   Experiments and results

4.1   Dataset

For the first experiment training, we used dataset "*A list of English positive and negative opinion words or sentiment words*" [3], which was compiled by authors Hu and Liu (2004). It is actually a list of opinion lexicon. List of positive words contains 2006 words and negative list contains 4783 words. We add additional column in this list, which named "Sentiment". This column contains two values: "positive" for positive words list and "negative" for negative words list. After we combine these two lists in to one. The prepared training dataset contains 6789 words.

The dataset for testing was acquired from Movie Review Data site of Cornell University Department of Computer Science, which was created by Pang and Lee, (2004) and contains 1000 positive and 1000 negative processed reviews (polarity dataset v2.0 [4]). Dataset was splitted into training data (70%), which was used in second experiment as the training dataset (1400 movie reviews) and testing data (30%). We used the same testing dataset (600 movie reviews) in first and the second experiments.

For the third experiment we used The Stanford Twitter sentiment corpus (sentiment140[5]) dataset, introduced by Go et al. (2009). The dataset contains 1.6 million tweets automatically labelled as positive or negative based on emotions. For our experiment we used 50000 positive and 50000 negative (total 100000) randomly selected tweets. The dataset was splitted into training (70%) and testing (30%) datasets.

For the fourth experiment we used Amazon customer reviews dataset [6]. The dataset contains 4 million reviews and star ratings. For our experiment we used 200000 positive and 200000 negative (total 400000) randomly selected reviews. The dataset was splitted into training (70%) and testing (30%) datasets.

4.2   Experimental settings

The proposed technique was implemented using R language (R Core Team, 2016), package e1071 (Meyer et al., 2017), which includes both SVM and Naïve Bayes implementations. SVM implementation is based on popular SVM package LibSVM; here, default *C*-classification technique which corresponds to the default SVM classifier was used. All parameters were set to their default values.

Default Naïve Bayes parameters:
- laplace (positive double controlling Laplace smoothing): 0 *(disables Laplace smoothing)*
- na.action (A function to specify the action to be taken if "*NA*" are found): *The default action is not to count "NA" for the computation of the probability factors*

---

[3] `https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html`

[4] `http://www.cs.cornell.edu/people/pabo/movie-review-data/`

[5] `http://help.sentiment140.com/`

[6] `https://www.kaggle.com/bittlingmayer/amazonreviews/`

Default SVM parameters:
- type: *C-classification*
- kernel: *linear*
- gamma: 1/*(data dimension)*
- cost of constraints violation (cost): 1
- tolerance of termination criterion (tolerance): 0.001
- epsilon in the insensitive-loss function (epsilon): 0.1

Also, the threshold values were selected by manually investigating the results. We found that the performance was optimal when word selection probability for was set to $p \geq 0.8$, therefore it was selected as threshold value th_1 for the algorithm for words. Similarly, $th_2 = 0.8$ threshold value was manually selected for the algorithm for sentences.

4.3 Results

Four experiments were executed to evaluate the performance of proposed techniques:

1. In first experiment we used training dataset, which contains a list of English positive and negative opinion words or sentiment words. List of positive words contains 2006 words and negative list contains 4783 words (Hu and Liu, 2004), in total 6789 words. For testing we used movie review dataset (Pang and Lee, 2004) which contains 300 positive and 300 negative movie reviews, resulting in total 600 movie reviews. The tokenized sentences were used as input for machine learning algorithms.
2. In second experiment we used the training movie review dataset (Pang and Lee, 2004) which contains 700 positive and 700 negative movie reviews, resulting in total 1400 movie reviews, and the same testing dataset from the first experiment to compare prediction efficiency between the tokenized sentences and the whole sentences.
3. In third experiment we used The Stanford Twitter sentiment corpus (sentiment140) dataset, introduced by Go et al. (2009). For our experiment we used 50000 positive and 50000 negative (total 100000) randomly selected tweets. The dataset was splitted into training (70%) and testing (30%) datasets.
4. Finally, in the last experiment we used Amazon customer reviews dataset. For our experiment we used 200000 positive and 200000 negative (total 400000) randomly selected reviews. The dataset was splitted into training (70%) and testing (30%) datasets.

Performance is measured using statistical measures: accuracy, precision, recall and F1 score. Formulas are presented below (Sammut and Webb, 2011):
Accuracy (ACC):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

Positive predictive value (PPV):

$$PPV = \frac{TP}{TP + FP}$$

Table 1. Experimental settings

Exp No.	Training features	Training dataset	Testing features	Testing dataset
1	words	6789	movie reviews	600
2	movie reviews (70%)	1400	movie reviews (30%)	600
3	tweets (70%)	70000	tweets (30%)	30000
4	Amazon Reviews (70%)	280000	Amazon Reviews (30%)	120000

Negative predictive value (NPV):

$$NPV = \frac{TN}{TN + FN}$$

True positive rate (TPR):

$$TPR = \frac{TP}{TP + FN}$$

True negative rate (TNR):

$$TNR = \frac{TN}{TN + FP}$$

Harmonic mean of PPV and TPR ($F_1 score$):

$$F_1 score = \frac{2}{\frac{1}{PPV} + \frac{1}{TPR}}$$

where *TP* - count of correctly classified "positive" sentiments, *TN* - count of correctly classified "negative" sentiments. *FP* - count of incorrectly classified "positive" sentiments. *FN* - count of incorrectly classified "negative" sentiments.

Table 2 shows that the best results we got in forth experiment when was recognizing Amazon reviews sentiments. Our introduced method gave accuracy (ACC) 89,19%, while accuracy of SVM (ACC) was 89,05% and Naïve Bayes (ACC) 84,35 %. Not far away from the best results is the second experiment when was recognizing movie reviews with accuracy (ACC): our introduced method 88,66 %, SVM 88,50 % and Naïve Bayes 81,67 %. The first experiment where was used a list of English positive and negative opinion words or sentiment "*A list of English positive and negative opinion words or sentiment words*" (Hu and Liu, 2004) for recognize movie reviews, showed the lowest recognize accuracy, but our introduced method still outperform SVM and Naïve Bayes and gave accuracy (ACC) 72,00%. To compare the first and the second experiments, where was used the same testing dataset, we found that the better accuracy is obtained when the sentences is not tokenized and the training dataset is from the same domain. In third experiment, when was recognizing sentiments from tweets, our introduced method shown accuracy (ACC) 78,31% and again outperform SVM with accuracy (ACC) 78,08 % and Naïve Bayes with accuracy 75,77 %. As we can see the best results are when recognizing movie reviews and Amazon reviews, the results of tweets recognizing is not very high, but still good enough if we don't need a very high

Table 2. Results

Classifiers	ACC	PPV	NPV	TPR	TNR	$F_1 score$
Experiment 1						
SVM	71,33%	64,33%	78,33%	74,81%	68,71%	69,16%
Naïve Bayes classification	67,83%	71,67%	64,00%	66,56%	69,31%	69,02%
Our introduced method	72,00%	65,67%	78,33%	75,19%	69,53%	70,11%
Experiment 2						
SVM	88,50%	89,67%	87,33%	87,62%	89,42%	88,63%
Naïve Bayes classification	81,67%	76,33%	87,00%	85,45%	78,61%	80,63%
Our introduced method	88,66%	90,00%	87,33%	87,66%	89,73%	88,82%
Experiment 3						
SVM	78,08%	78,74%	77,43%	77,72%	78,46%	78,23%
Naïve Bayes classification	75,77%	70,75%	80,79%	78,65%	73,42%	74,49%
Our introduced method	78,31%	77,82%	78,80%	78,59%	78,04%	78,20%
Experiment 4						
SVM	89,05%	88,11%	89,99%	89,80%	88,33%	88,95%
Naïve Bayes classification	84,35%	79,96%	88,75%	87,66%	81,58%	83,64%
Our introduced method	89,19%	88,10%	90,32%	90,10%	88,33%	89,07%

accuracy. This happen because tweets are very short and tweets contain noises, slangs, acronyms and etc.

To compare with SVM and Naïve Bayes classification, our introduced method provided more uniform recognition of both classes (exept the first experiment where we gave almost the same), compared to other methods. $PPV, NPV, TPR, TNR, F_1 score$, are almost even in our introduced method, while Naïve Bayes have spread from 76,33% till 87,00% in second experiment, from 70,75% till 80,79% in third experiment and from 79,96% till 88,75% in fourth experiment. It can be indicated that Naïve Bayes classifier performed weekly in all experiments, but its combination with stronger classifier, such as SVM, can improve performance of the latter.

Results suggest that training and testing datasets should come from the same domain, which limits the direct transfer of the pretrained classifier to other domains. Also, Naïve Bayes classifier did not perform well while recognizing sentiments in all experiments to compare with SVM and our introduced method..

5 Conclusions and future work

The main idea of this paper was introduce and to test a new method with datasets which are from different areas and have different size. We compared two supervised machine learning algorithms of SVM and Naïve Bayes classification with our introduced method for the movie reviews, tweets and Amazon reviews sentiment recognizing.

The experimental results show that our introduced method outperform SVM and Naïve Bayes classification in all experiments. The different datasets and size of datasets were selected to check if our method could be applied in different areas and still can

give the better accuracy. It was tested: on a huge text dataset like movie reviews, which contains long text and several or more sentences; on a short text dataset like tweets, which contains mostly 140 symbols; on a different size text like Amazon reviews. New method outperform the SVM from (ACC) 0,14% till 0.67% and Naïve Bayes (ACC) from 2,84% till 6,99% . This is not very high in case of SVM, but it is very important if we need more accuracy. The main advantage is that our introduced method provided more uniform recognition of both classes $PPV, NPV, TPR, TNR, F_1 score$. The values between them are almost equal, to compare with Naïve Bayes which has significant differences.

The accuracy we got in our approach is enough for movie reviews, tweets and for Amazon reviews. Also the better accuracy is obtained when the sentences is not tokenized. In our paper, only the texts from the tweets are used and other information is cleaned. This approach is applicable in different areas, but the parameters could be tuned, and classifier must be developed using datasets from particular domain.

Sentiment analysis is very challenging area. Text in messages can contain sarcasm, linguistic issues, emoticons, spam and etc. We need to understand the whole context of the sentence, because even one word can change polarity of sentence. In the future work, we will work on this issue, will try to increase accuracy of classification by tuning algorithms parameters and optimize machine learning algorithms.

References

Barbosa, L., Feng, J. (2010). Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, 36–44.

Boser, B.E., Guyon, I.M., Vapnik, V.N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, 144–152.

Catal, C., Nangir, M. (2017). A sentiment classification model based on multiple classifiers. *Applied Soft Computing, 50*, 135–141.

Chang, C.C., Lin, C.J. (2011). LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3), 27.

Cortes, C., Vapnik, V. (1995). Support-vector networks. *Machine learning, 20*(3), 273–297.

Dave, K., Lawrence, S., Pennock, D.M. (2003, May). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, 519–528.

Davidov, D., Tsur, O., Rappoport, A. (2010). Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd international conference on computational linguistics: posters*, 241–249.

Engonopoulos, N., Lazaridou, A., Paliouras, G., Chandrinos, K., (2011). ELS: a word-level method for entity-level sentiment analysis. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*.

Farra, N., Challita, E., Assi, R.A., Hajj, H. (2010). Sentence-level and document-level sentiment mining for arabic texts. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference*, 1114–1119.

Go, A., Bhayani, R., Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford, 1*(2009), 12.

Hu, M., Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168–177.

Hussein, D.M.E.D.M. (2016). A survey on sentiment analysis challenges. *Journal of King Saud University-Engineering Sciences*.

Khairnar, J., Kinikar, M. (2013). Machine learning algorithms for opinion mining and sentiment classification. *International Journal of Scientific and Research Publications*, 3(6), 1–6.

Kharde, V., Sonawane, P. (2016). Sentiment analysis of twitter data: A survey of techniques. *arXiv preprint arXiv:1601.06971.*

Le, B., Nguyen, H. (2015). Twitter sentiment analysis using machine learning techniques. In *Advanced Computational Methods for Knowledge Engineering*, 279–289, Springer, Cham.

Liu, B. (2010). Sentiment Analysis and Subjectivity. *Handbook of natural language processing*, 2, 627–666.

Manek, A.S., Shenoy, P.D., Mohan, M.C., Venugopal, K.R. (2017). Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier. *World wide web, 20*(2), 135–154.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F. (2017). e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.6-8. `https://CRAN.R-project.org/package=e1071`

Pang, B., Lee, L., Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume* 10, 79–86.

Pang, B., Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics* (p. 271). Association for Computational Linguistics.

Pang, B., Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(12), 1–135.

Sammut, C., Webb, G.I.(Eds.). (2011). *Encyclopedia of machine learning*. Springer Science & Business Media.

Tang, H., Tan, S., Cheng, X. (2009). A survey on sentiment detection of reviews. *Expert Systems with Applications*, 36(7), 10760–10773.

Team, R.C. (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. `https://www.R-project.org`

Thomas, B. (2013). What Consumers Think About Brands on Social Media, and What Bunesses Need to do About it. *Report, Keep Social Honest*.

Wan, Y., Gao, Q. (2015. An ensemble sentiment classification system of twitter data for airline services analysis. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*, 1318–1325.

Yessenalina, A., Yue, Y., Cardie, C. (2010). Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 1046–1056. Association for Computational Linguistics.

Zhou, H., Song, F. (2015). Aspect-Level Sentiment Analysis Based on a Generalized Probabilistic Topic and Syntax Model. In *FLAIRS Conference*, 241–245.