ISSN: 2229-6956 (ONLINE)
DOI: 0.21917/ijsc.2016.0175

AN ENSEMBLE APPROACH FOR SENTIMENT CLASSIFICATION: VOTING FOR CLASSES AND AGAINST THEM

T. Subbulakshmi[1] and R. Regin Raja[2]
School of Computing Science and Engineering, VIT University Chennai Campus, India
E-mail: [1]research.subbulakshmi@gmail.com, [2]reginsekar@gmail.com

Abstract

Sentiment denotes a person's opinion or feeling towards a subject that they are discussing about in that conversation. This has been one of the most researched and industrially promising fields in natural language processing. There are several methods employed for performing sentiment analytics. Since this classification problem involves natural language processing, every solution has its own advantages and disadvantages. Hence mostly, a combination of these methods provides better results. Various such ensemble approaches exist. The objective of this work is to design a better ensemble approach that uses a complex voting method, where classifiers are given rights not only to vote in favour of classes but also against them. This in turn will give chances to the algorithms that are weaker in classifying a sentence toward a particular class but better at rejecting it. The performance of the ensemble is compared to the individual classifiers used in the ensemble and also the other simple voting ensemble methods to verify whether the performance is better compared to them. The designed ensemble is currently implemented for sentiment analytics. This can also be used for other classification problems, where generalization is required for better results.

Keywords:

Sentiment Analytics, Ensemble Method, Sensitivity, Specificity

1. INTRODUCTION

Sentiment analytics is a classification problem in which the input provided will mostly be in text format, and the task is to classify them in order to provide the sentiment the input text holds. The basic classes normally used would be positive and negative classes. This can be further expanded by adding a class called neural since not all sentences are subjective. There also exist classifiers that use other types of sentiment measures such as emotional state like anger, sadness, happiness etc.

It provides very useful insights, thus gaining a lot of attention in both research and business industries. There are a lot of techniques that are being used which will be given a brief overview in this section along with their advantages and disadvantages.

Initially developed methods relied on dictionaries. Dictionaries had a set of words tagged with positive and negative sentiment. The sentences are evaluated for sentiment using the frequency of occurrences of the words from dictionary. Advantage of such a system is that, with good dictionaries and approach they can produce reliably good results. But these systems cannot handle complex problems that arise when it comes to text analytics such as psychological impact of the user, as they follow a simple word presence based approach. Also if the approach is made more complex in nature their speed drops considerably.

Methods using machine learning also exists. Since sentiment analytics can be regarded as a classification problem, machine learning can be used to find hidden patterns in such data and classify them. It basically involves extracting features from the text and training a model from a set of pre-classified data and using the model obtained to classify the data in the future. Some of the methods not only regard the text but also the other features obtained from the source such as author, country, date and time etc. Advantage of such method is that it captures the semantic structural differences in the classes and hidden relationship between features that are not derived from the text. On the other hand this system only works for the patterns that it has captured, whereas human conversations vary by several factors, thus having a wide variety of patterns in them.

Due to the varying characteristics of the classifiers mentioned above, ensemble algorithm which combines the result from various classifiers is most commonly used. This method helps in giving a more generalised result thus having a good accuracy compared to individual classifiers. There are ensembles that are built for machine learning on same set of training data and other kind of ensemble where the basic functionality of the individual classifiers itself widely vary from each other. For the latter case, many algorithms exist to process the results from the algorithms present in the ensemble. The basic method is voting based method where every algorithm results are considered as a vote in favour of the particular class they classified the input as. The class with the highest number of votes is chosen as the final result. Also there are methods where individual classifiers are given rights to vote based on weights. The method discussed in this paper is similar to this method but the rights are given to the algorithms not only to vote in favour of the class but also against them. Even though there are more complex methods, this paper discusses about this particular ensemble and compares it to the method it is derived from to show that it is better than its predecessor.

2. RELATED WORKS

Most commonly used ensemble in machine learning are bagging [1] and boosting [2]. They use a same set of training data and build various models to the ensemble then combine them. But if the component classifiers are of completely different methodologies, then very optimally suitable method is majority voting classification [3]. These basically work by adding a vote to the class if a classifier classifies the sentence as such. Voting can be done in various ways. Methods used in [4] and [5] use weighted approach for voting. In [4] the ensemble method used creates weights based on harmonic mean of precision and recall. Method discussed in [5] creates a voting vector based on whether the classifier has rights to vote or not. It is weighted based on a genetic algorithm. Similar to this, the proposed method also relies on creating voting vector for each classifier but also considers votes against the class. The proposed method uses only the binary

weights selected using certain threshold conditions to get the result.

Many methods are available for sentiment analytics. Methods which have their implementations available openly in R packages are mainly used in this study. Out of the methods studied, there are dictionary based works with different approaches, which are aimed for different purposes. In the work done by Liu et al. [6] products are compared based on their sentiment. The prominent features are extracted and the words associated with them are compared. Saif and Peter's work [7] apart from polar sentiment also identifies the emotional value of the sentence. Finn's method [8] discusses using a specific set of word list for micro blogging websites. Hu and Liu's method [9] extracts specific sentences identified as subjective sentences for opinion analysis then performs the analysis. Also there are machine learning methods which usually extract features as document-word vectors and may use other features. Some are discussed in [4]. Most prominently featured algorithm of machine learning is SVM described in the works such as [15], [13] and [14]. These algorithms usually use unigram and bigram as features for the classifiers. The sentiment analysis method described in "The Stanford CoreNLP Natural Language Processing Toolkit" by Manning et al. [11] describes about the tool called 'CoreNLP' created at Stanford which also can be used for various natural language processing tasks. This method uses a deep learning classifier which is trained by parse trees of sentences whose every subunit is tagged with sentiment.

Research in micro blogging websites like twitter, is becoming well known as it is a very recent form of communication and holds a lot of information even in a short span of time. Initial works of Alec et al. [13] involved testing various machine learning approaches on these data. Work of Kouloumpis et al. [14] focuses on the new features to consider while classifying tweets. Twitter provides a large number of other metadata such as user id, country, number of re-tweets, time zone, time of tweet, etc. Barbosa and Feng [16] suggested using these features, since twitter has a large vocabulary and short texts which traditional approaches are not well suited for.

3. PROPOSED METHOD

Usually the ensemble logic allows classifiers to vote in favour of classes alone. The system proposed here also focuses on allowing the algorithms to vote against them. Thus both the factors are considered while deriving at the final conclusion. To prove that this approach performs better than the other voting methods this has been implemented for sentiment analytics. First the ensemble has to be created and then use it to classify.

3.1 METHOD OF CREATING THE ENSEMBLE

Following are the steps followed in creating the ensemble.

3.1.1 Collecting Various Algorithms:

Algorithms that are widely used for the same problem, in this case sentiment analytics, are chosen.

3.1.2 Modifying Them to Have a Standard Format:

The classification algorithms should be modified to produce a standard set of result classes, for the comparison to be easier. In case of sentiment analytics, the result classes should either be positive, negative or neutral.

3.1.3 Calculating Sensitivity and Specificity:

The results were collected for algorithms, tested for manually classified data points. Confusion matrices should be constructed for all classifiers. Based on this result, sensitivity and specificity should be measured for the classifiers. The Table.1 shows the basic structural format of the confusion matrix,

Table.1. Confusion matrix

	Predicted Class		
Actual Class		**Yes**	**No**
	Yes	True Positive (A)	False Negative (B)
	No	False Positive (C)	True Negative (D)

The sensitivity and specificity are calculated based on the data from the Table.1 using the following formulae,

$$Sensitivity = \frac{A}{A+B} \quad (1)$$

$$Specificity = \frac{D}{C+D}. \quad (2)$$

3.1.4 Creating Voting Vector:

After calculating them, a voting vector is chosen for every classifier. The vector consists of the following pair for each class,

- Belongs to the class
- Does not belong to the class

The Table.2 shows the voting vector for sentiment analytics. Sensitivity is used as a threshold for voting the class and specificity is used as a threshold for voting against the class.

Table.2. Voting Vector format

Is Positive Class	Is Not Positive Class	Is Negative Class	Is Not Negative Class	Is Neutral Class	Is Not Neutral Class

3.2 METHOD BY WHICH THE ENSEMBLE IS USED FOR CLASSIFICATION

After the ensemble is constructed it must be able to classify newly available data. Hence the approach as shown in the Fig.1 is used. It is explained as follows,

3.2.1 Data Cleaning:

Mostly available text data that need sentiment analytics are social media data. These texts are a mixture of words, links, user names, hash tags, emoticons and other data that are irrelevant to the classifier being used. Hence each individual classifier may have to use different approach for data cleaning. Once the data has been cleaned, it is then sent to the classifiers.

3.2.2 Classifying:

Each classifier classifies the data then produce results. These results are converted based on the standardised result format.

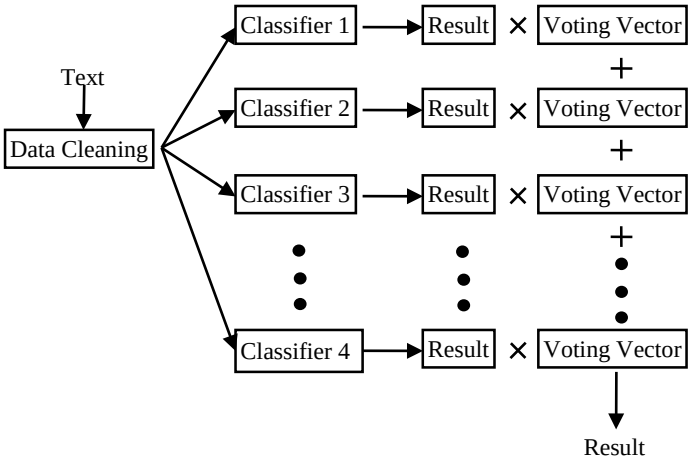

Fig.1. Ensemble Method based on Voting

3.2.3 Gathering of Results:

The results are collected and then are converted to vectors. The format is as shown in the Table.2. '100101'is the vector for positive class, '011001' is for negative class and '010110' is for neutral class. The results are finally stored in the form of classifier to voting vector matrix.

3.2.4 Applying Voting Vector:

Voting vectors were constructed for individual classifiers already while building the ensemble method. These vectors are then multiplied with the result matrix. The result obtained will be the votes for individual classifiers that it can perform based on the rights it received and the class it has chosen.

3.2.5 Finalizing the result:

Once the voting is done, then the result is added along the columns. Thus we get the result as the number of votes for every category. Each class will have votes in favour and against them. Their ratio gives the final feedback for each class. The class which has maximum value for this ratio is chosen as the final result of the ensemble.

4. IMPLEMENTATION

Individual classifiers of the ensemble were chosen based on their performance in a test data and the ensemble was built. The process is clearly described in this section.

4.1 CLASSIFIERS USED

The sentiment classifiers that are available in CRAN's repositories were studied. Totally five classifiers were chosen for this study. Their results vary based on the purposes they were designed for. To make them return common classes of outputs, the chosen classes were positive, negative and neutral. This process is required in order to make the approach more general result. Also adding new classifiers to the ensemble will be much simpler.

4.1.1 QDAP:

'Qdap' package of R provides a lot of useful text mining tools. One of its function 'polarity' is used to return sentiment value. This is a dictionary based classifier based on the dictionary from the work of Hu and Liu [9]. It also has an improved multiple dictionary referencing system. It allows referencing of external dictionaries to match the domain it is being used in. This algorithm will hence forth will be referred to as 'qdap'. It provides the following dictionaries that can be modified by the user if necessary,

- **'Words'** is a dictionary of positive and negative words.
- **'Amplifiers'** is a dictionary that contains words that strengthens the effect of the positive or negative word which comes along it.
- **'De-amplifiers'** is a dictionary that contains words that weakens the effect of positive or negative word which comes along it.
- **'Negators'** is a dictionary that contains words that reverses the polarity of the polar words that comes along it.

4.1.2 NRC:

The 'syuzhet' package in R provides a function called 'get-nrc-sentiment' which is designed based on the work of Saif and Peter [7]. This algorithm will hence forth be referred to as 'nrc'. The dictionary used consists of words and phrases tagged with emotional value such as anger, anticipation, disgust, fear, joy, sadness, surprise and trust. It also provides polarity of the sentence. This polarity value is used to return the sentiment value required for the ensemble.

4.1.3 AFINN:

The package 'syuzhet' of R also provides a method called 'get-sentiment' for which we can choose the method of approach. On choosing 'afinn' as the approach, sentiment is calculated based on the work of Finn [8]. This algorithm will hence forth will be referred to as 'afinn'. It classifies sentences based on a dictionary which has words and phrases tagged with sentiment by Finn Arup mainly focusing on micro blogging.

4.1.4 Bing:

'Bing' is also a method of sentiment classification provided by 'find-sentiment' method of 'syuzhet' package. It is designed based on the work Hu and Liu in [6] and [10]. This algorithm will hence forth be referred to as 'bing'. They have designed an approach focusing on classifying reviews.

4.1.5 CoreNLP:

R's package 'coreNLP' provides an interface to a lot of natural language processing implementations of Stanford's coreNLP [11]. The method 'getSentiment' returns the sentiment classified based on the Stanford's implementation of sentiment analytics in coreNLP. The classifier is trained by a deep learning algorithm using parsed trees as training set. This algorithm will hence forth be referred to as 'corenlp'. Before classifying, the sentences are annotated with parts of speech, sentence split and converted to parse trees. Then the model classifies it.

4.2 DATA USED

The data set used for testing these classifiers is the Sentiment140's manually classified test set which is described in [12]. The data consists of manually classified tweets with the distribution as shown in the Table.3.

Table.3. Class Distribution in the data used

Class	Count
Positive	182
Negative	177
Neutral	139
Total	498

4.3 BUILDING THE ENSEMBLE CLASSIFIER

The confusion matrix is constructed for every classifier discussed in section 4.1 using the data described in section 4.2. The sensitivity and specificity measures are measured for the classifiers and are listed in the Table.2 and Table.3.

Table.4. Sensitivity measures of classifiers

Method	Sensitivity		
	Positive	Neutral	Negative
Qdap	0.6374	0.7482	0.6102
Afinn	0.7198	0.7266	0.5706
Bing	0.6648	0.7698	0.5367
Nrc	0.5714	0.5971	0.3616
Corenlp	0.3462	0.3669	0.6949

Table.5. Specificity measures of the classifiers

Method	Specificity		
	Positive	Neutral	Negative
qdap	0.8576	0.7298	0.9128
afinn	0.8070	0.7827	0.9190
bing	0.8513	0.6908	0.947
nrc	0.7753	0.6017	0.8972
corenlp	0.9367	0.7716	0.5047

In the ensemble, for the case of sentiment classification, the voting vector has six digits with format as shown in Table.2. For example the voting vector for coreNLP was 011001, hence it can vote in favour of negative class and against positive and neutral classes.

Table.6. Finalized Voting Rights matrix for the classifiers

	Positive		Negative		Neutral	
	Yes	No	Yes	No	Yes	No
Qdap	1	1	1	1	1	1
Nrc	0	0	0	0	0	0
Afinn	1	1	0	1	1	1
Bing	1	0	1	1	1	0
Corenlp	0	1	1	0	0	1

After conducting a lot of experiments, the threshold set in the current approach is that a classifier is allowed to vote in each of the six divisions if it belongs to the top three classifiers of the particular division. The measures used are sensitivity to vote in favour of the class and specificity to vote against it. Thus based on the results from the Table.4 and Table.5 each method was given rights to vote. The finalized voting rights for the classifiers can be seen through the classifier to voting-vector matrix shown in the Table.6. From the table we can observe that the best ensemble does not include the method 'nrc' for any voting. This is because 'nrc' is not present in top three of any of the categories. This may be because 'nrc', a method which uses emotional lexicon, may not be good for the data used i.e. short tweets.

5. EXPERIMENTS AND RESULTS

The objective of the experiments conducted is to measure the accuracy difference between various ensemble approaches. Hence the focus was not in improving the accuracy of individual classifiers but to compare the various threshold measures that can be used with the proposed approach and also compare different ensemble approaches.

In order to compare the performance of the classifiers, the accuracies were measured for individual classifiers and also the ensemble methods. The data used for this is the test set from Sentiment 140 [12]. The measured accuracies for individual classifiers are displayed in the Fig.2.

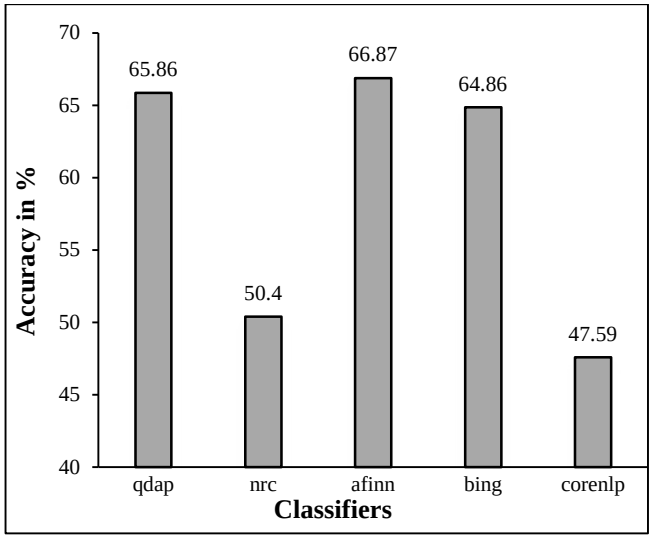

Fig.2. Bar-Graph: Accuracies of the Individual Classifiers

As we can see, the classifiers are showing different accuracies. This is because the classifiers have been designed with different purposes or goals.

Before comparing the ensemble approach to other approaches, some experiments were made to tune the classifier. The classifier was tuned by setting certain thresholds which had to be considered to make the classifier fair and accurate.

5.1 CHOOSING THE THRESHOLD

The voting rights must be given to the classifiers based on some thresholds without any bias. Two thresholds were tested in the process of creating the voting vector for the classifiers. One of

the thresholds is the number of best classifiers that are to be allowed in a single category to vote. If this is not set then the ensemble may end up with too many classifiers to vote for some categories, while no or few classifiers to vote for some categories. The second threshold is to allow the classifiers with sensitivity and specificity above a certain base value to vote for the respective category. This threshold is set to allow only the strong classifiers to vote. In the next two sections these thresholds are discussed with more details along with the experiments conducted to select the best among them.

5.1.1 Based on the Number of Votes in Each Category:

In this experiment the number of votes per category was increased from 1 to 5, as there are only five classifiers, and the resulting ensemble's performance was measured. Their values are shown in the Fig.3. From the graph we can see that the accuracies are more or less same. But also we can see that allowing all the classifiers to vote will comparatively reduce the performance. Also allowing three classifiers to vote per category has the highest accuracy. So here after this threshold value will be 3 classifiers per category.

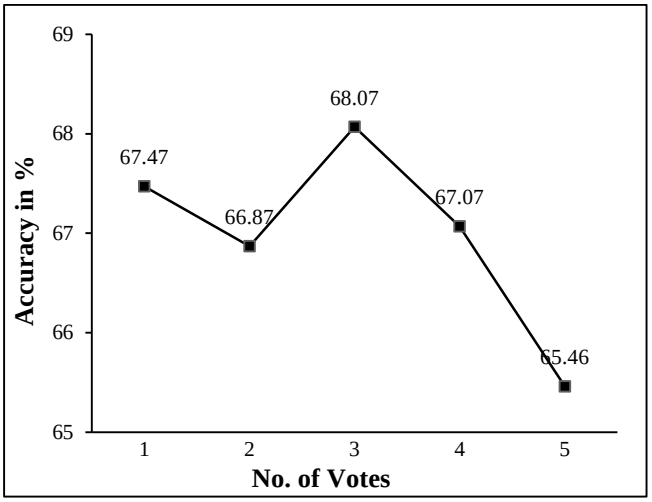

Fig.3. Line-Graph: Accuracies Based on Votes Allowed

5.1.2 Based on a Base Value for Sensitivity and Specificity:

This experiment was conducted to verify whether setting a base value for sensitivity and specificity as threshold is better than setting equal rights. Both the methods have their own advantage and disadvantage. Equal voting reduces the bias towards the category, but it may lead to choosing a low performing classifier for a category. On the other hand using a base value constricts the ensemble from using bad classifiers, but it may lead to having bias towards a particular category. Hence both the methods are compared. Their best results can be compared using the Fig.4.

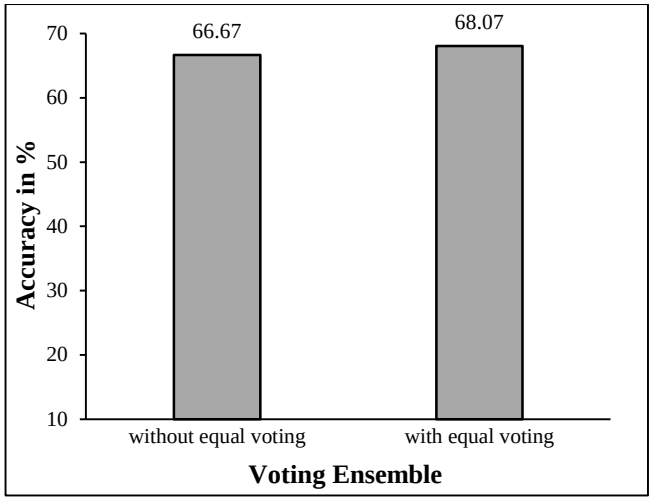

Fig.4. Bar-Graph: Accuracies based on different threshold measures

The category without equal voting is based on setting the base value of 0.6 in both sensitivity and specificity. This made only good classifiers to participate in voting but it became highly biased as the number of classifiers to vote in negative class was less. The equal voting method is more generalized in this sense hence this ensemble has been chosen for further comparison.

5.2 COMPARISION WITH OTHER APPROACHES

The ensemble approach designed is compared with three other approaches. All of these approaches have voting only in favour of classes. They do not consider voting against the classes.

5.2.1 Allowing All the Classifiers to Vote:

In this ensemble approach, all the classifiers are given equal rights to vote for the classes. Thus making it the simplest approach of all the ensembles compared.

5.2.2 Allowing only Selective Classifiers to Vote:

An improved approach will be to allow only the classifiers that had a better performance with the test set for the particular class to vote. The classifiers were first chosen by letting them classify a small test set. A test set must contain the data points or sentences which are specific to the domain where the classifier would be used. A threshold will be set to accept the classifiers that pass it. So this will omit the classifiers that were not designed for the same goal. The remaining classifiers are chosen for classification.

5.2.3 Voting Based on Weights:

In this ensemble approach, a weight is assigned to every classifier based on the averaged performance measure. Then these weights are taken as the votes rather than binary votes. Thus good classifiers get to have good impact on the result.

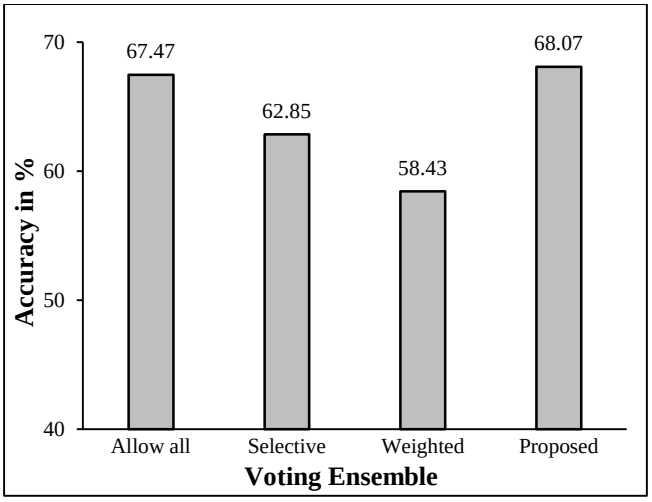

Fig.5. Bar-graph: Accuracies of the Ensemble Methods

The result from the graph in Fig.5 shows the difference between the different ensembles discussed above. The weighted voting method performance is affected by high specificity measure of the low performing classifiers, resulting in low performance. Its performance may improve if the classifiers used were strong individually. Also compared to the basic voting method the selective method should have performed better but it didn't. This may be due to several reasons. Particularly to this case, it is because the number of classifiers significantly reduces since there were only five classifiers that were used initially for selection process. The proposed method performs better even with the above constraints holding the other ensembles from performing better.

As the ensemble method which includes disapproval votes is more generalized than other methods, its performance is better compared to the individual classifiers and the other ensembles.

6. CONCLUSION

Five classifiers which were available in R were chosen. An improved voting ensemble algorithm was designed. Then the classifiers were given rights to vote in favour of the class and to vote against the class. Due to this even weak classifiers with less accuracies like 'coreNLP' were found to have better impact on the final result if they have good accuracy in voting against a class. This helped in generalizing the result of individual classifiers. The threshold for getting voting rights was chosen through various experiments. Also this generalization showed good result when tested and compared to the results of other voting ensemble methods which focus only on voting in favour of the classes. Since the ensemble works only after the results are gathered from the individual classifiers whose results have been standardised, this approach is also suitable for other classification problems.

REFERENCES

[1] Leo Breiman, "*Bagging Predictors*", Technical Report, Department of Statistics, University of California, pp. 1-19, 1994.

[2] Robert E. Schapire, "The Boosting Approach to Machine Learning: An Overview", *Nonlinear Estimation and Classification*, Vol. 171, pp. 141-171, 2003.

[3] Gareth James, "Majority Vote Classifiers: Theory and Applications", Ph.D Dissertation, Department of Statistics, Stanford University, pp. 1-123, 1998.

[4] Pollyanna Goncalves, Matheus Arjujo, Fabricio Benevenuto and Meeyoung Cha, "Comparing and Combining Sentiment Analysis Methods", Proceedings of 1st ACM Conference on Online Social Networks, pp. 27-38, 2013.

[5] Sriparna Saha and Asif Ekbal, "Combining Multiple Classifiers using Vote Based Classifier Ensemble Technique for Named Entity Recognition", *Data and Knowledge Engineering*, Vol. 85, pp. 15-39, 2013.

[6] Bing Liu, Minqing Hu and Junsheng Cheng, "Opinion Observer: Analyzing and Comparing Opinions on the Web", *Proceedings of 14th International World Wide Web Conference*, pp. 10-14, 2005.

[7] Saif Mohammad and Peter Turney, "Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon", *Proceedings of Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pp. 26-34, 2010.

[8] Finn Arup Nielsen, "A New Anew: Evaluation of a Word List for Sentiment Analysis in Microblogs", *Proceedings of Workshop on Making Sense of Microposts: Big Things Come in Small Packages*, pp. 93-98, 2011.

[9] Minging Hu and Bing Liu, "Mining Opinion Features in Customer Reviews", *Proceedings of 19th National Conference on Artificial Intelligence*, pp. 755-760, 2004.

[10] Minqing Hu and Bing Liu. "*Mining and Summarizing Customer Review*", *Proceedings of 10th ACM International Conference on Knowledge Discovery and Data Mining*, pp. 168-177, 2004.

[11] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard and David McClosky, "The Stanford Core NLP Natural Language Processing Toolkit", *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60, 2014.

[12] Alec Go, Richa Bhayani and Lei Huang, "Twitter Sentiment Classification using Distant Supervision", Technical Report CS224N, Stanford University, pp. 49-54, 2009.

[13] Alec Go, Lei Huang and Richa Bhayani, "Twitter Sentiment Analysis", Final Project Report, Stanford University, pp. 1-16, 2009.

[14] Efthymios Kouloumpis, Theresa Wilson and Johanna Moore, "Twitter Sentiment Analysis: The Good the Bad and the OMG!", *Proceedings of 5th International Association for the Advancement of Artificial Intelligence Conference on Weblogs and Social Media*, pp. 538-541, 2011.

[15] Bo Pang, Lillian Lee and Shivakumar Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques", *Proceedings of 2nd Conference on Empirical Methods in Natural Language Processing*, Vol. 10, pp. 79-86, 2002.

[16] Luciano Barbosa and Julan Freng, "Robust Sentiment Detection on Twitter from Biased and Noisy Data", *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 36-44, 2010.