

Automatic Sentiment Analysis of Twitter Messages Using Lexicon Based Approach and Naive Bayes Classifier with Interpretation of Sentiment Variation

Pravin Keshav Patil¹, K. P. Adhiya²

P.G. Student, Dept. of Computer Engineering, SSBT College of Engineering and Technology, Bambhori, Jalgaon, Maharashtra, India¹

Associate Professor, Dept. of Computer Engineering, SSBT College of Engineering and Technology, Bambhori, Jalgaon, Maharashtra, India²

ABSTRACT: With the explosive growth of user generated messages, twitter has become a social site where millions of users can exchange their opinion. Sentiment analysis on twitter data has provided an economical and effective way to expose public opinion timely, which is critical for decision making in various domains. For instance, a company can study the public sentiment in tweets to obtain users' feedback towards its products; while a politician can adjust his/her position with respect to the sentiment change of the public. There have been a large number of research studies and industrial applications in the area of public sentiment tracking and modelling. Millions of users share their opinions on Twitter, making it a valuable platform for tracking and analyzing public sentiment. Such tracking and analysis can provide critical information for decision making in various domains. Therefore it has attracted attention in both academia and industry. Previous researches showed that the tweet was classified appropriately only if the tweet would contain the exact same label (use to detect sentiment) as in the training set. But this approach fails when the tweet contains a synonym or a variant of the label (having same meaning) instead of the exact same label. Although the tweet should have been classified accurately because the variant in the tweet and the label in the training set had same meaning. To solve this problem, a Lexicon based approach using naive bayes classifier for automatic analysis of twitter message is presented. The results show that by incorporating a lexicon based approach with the bayes classifier, the efficiency and the accuracy of the classifier to classify the tweets has improved significantly.

KEYWORDS: Naive Bayes classifier, Opinion Mining, Lexicon, Sentiment Analysis, Twitter

I. INTRODUCTION

Before the web, when an individual needed to make a decision, he/she typically asked for opinions from friends and families. When an organization wanted to find the opinions or sentiments of the general public about its products and services, it conducted opinion polls, surveys, and focus groups. In many cases, opinions are hidden in long forum posts and blogs. It is difficult for a human reader to find relevant sources, extract related sentences with opinions, read them, summarize them, and organize them into usable forms. Thus, automated opinion discovery and summarization systems are needed. Sentiment analysis, grows out of this need. Today Twitter has become the most popular microblogging service in social networking. Invented in 2006, Twitter was designed for communicating using short messages. This simplified communication and connection accelerated the process of message update. Today over 500 million tweets are being posted daily. This has made twitter an important repository for data analysis and mining of opinion, moods and views of socially interested people. Twitter has become an important place for most of the companies to promote their brands. Monitoring of these messages helps them to understand the constant changing needs of people and views

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2015

about products features and services. Today many reality TV shows use these tweets to get the feed back. Sentiment analysis is the process of determining the sentiments that the writer intended to transmit in his/her messages. This sentiment normally represents text polarity i.e. whether the opinion is positive or negative. Sentiment analysis is the concept under the data mining, where it is a resulting technique for extracting, classifying, perceptive and assessing the sentiments spoken in the different websites, social media insides and other user generated context.

II. RELATED WORK

Raymond Kosala, et.al., in [1] surveyed the research in the area of web mining. It also explores the connection between the web mining categories and the related agent paradigm. For this survey, focus is on the representation issues, on the process, on the learning algorithm and on the application of the recent works as the criteria. It describes the research done for the information retrieval giving an IR view of the unstructured documents. Also, the information retrieval view for semi-structured documents is discussed. The database view for the web content has been explained in detail which mainly tries to model the data on the web and to integrate them so that sophisticated queries other than keywords based search could be performed.

Sandra Stendahl, et.al., in [2] focused on different implementations on web mining and the importance of filtering out calls made from robots to get knowledge about the actual human usage of a website. This is to find patterns between different web pages and create more customized and accessible web pages to users, which in turn creates more traffic and trade to the website. Also, some common methods to find and eliminate the web usage made from robots while keeping browsing data made from human users intact are addressed. Web mining is viewed as seen to consist of three major parts: collecting the data, preprocessing the data and extracting and analyzing patterns in the data.

D. Jayalatchumy, et.al., in [3] worked on survey on the existing techniques of web mining and the issues related to it. It primarily reports the summary of various techniques of web mining approached from the following angles like feature extraction, transformation and representation and data mining techniques in various application domains. The survey on data mining technique is made with respect to Clustering, classification, sequence pattern mining, association rule mining and visualization. It also gives the overview of development in research of web mining and some important research issues related to it. It describes the process of web cleaning which is needed to remove noise and correct inconsistencies in the data.

Abdelhakim Herrouz, et.al., in [4] discussed the overview of different web content mining tools. Web content mining is the process of extracting useful information from the web documents. With the flood of information and data on the Web, the content mining tools helps to download the essential information that one would require. R.Malarvizhi, et.al., in [5], made a comprehensive study of the various web content mining techniques tools & algorithms. Arvind Arasu, et.al., in [6], proposed an algorithm for structure data extraction. Michael J. Cafarella, et.al., in [7], described three recent extraction systems that can be operated on the entire web. Nilesh Dalvi, et.al., in [8] analysed the nature and distribution of structured data on the web.

Miloš Radovanovic, et.al., in [9] discussed various approaches and applications of text mining. Ian H. Witten in [10] primarily focused on enhancing the relationship between text mining and data mining. Vishal Gupta, et.al., in [11] made a survey about different text mining techniques and applications. Xiaowen Ding, et.al., in [12] discussed the study of problem of determining the semantic orientations (positive, negative or neutral) of opinions expressed on product features in reviews. Ion Smeureanu, et.al., in [13] described supervised opinion mining techniques on online user reviews. The proposed algorithm in detects sentiments on movie user reviews, based on NB classifier.

Growing number of recent studies have been focused on the opinion mining, K.Nathiya, et.al., in [14] proposed a modified algorithm which mainly focuses on the experts opinions so that the readers could understand the content easily. Amandeep Kaur, et.al., in [15] made a survey on work done by different researchers on sentiment analysis and opinion mining techniques.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2015

Arti Buche, et.al., in [16] made a survey on different techniques for opinion mining and analysis.

Analysing this survey, Poongodi S, et.al., in [17] made a comparative study on different classification techniques for opinion mining. One of the important aspects in opinion mining is Feature Extraction. Feature extraction in opinion mining can be done by various ways like that of clustering, support vector machines etc. Pravesh Kumar Singh in [18] discussed the techniques of feature extraction for opinion mining.

Ayesha Rashid, et.al., in [19] surveyed the techniques, applications, long and short future research areas, Research gaps and future challenges in opinion mining. Hemalatha, et.al., in [20] came up with a sentiment analysis tool that used three machine learning algorithms, NB, SVM and MaxEnt. One advantage of MaxEnt is its flexibility which comes at the cost of large computational complexities.

Ana CE S Lima, et.al., in [21] described the automatic sentiment analysis for twitter messages. It covered three approaches for sentiment classification, Word based, emoticon based and hybrid approach. S.L. Ting, et.al., in [22] studied the used of NB classifier for document classification. Hiroshi Shimodaira, in [23], described text classification using the Naive Byes. NB provides two variants for text classification, one is the bernoulli document model and second is the the multinomial document model. In Bernoulli document model: a document is represented by a binary feature vector, whose elements indicate absence or presence of corresponding word in the document. In Multinomial document model a document is represented by an integer feature vector, whose elements indicate frequency of corresponding word in the document.

Alec Go, et.al., in [24] described about how to extract tweets from the twitter using the twitter API. The twitter provides a Twitter API which can be used to extract tweets using a query term. The tweets can also be extracted using the emoticons. Also it is observed that machine learning algorithms (NB) have accuracy above 80 percent when trained with emoticon data. Alexander Hogenboom, et.al., in [25] described the use of emoticons to convey the sentiments and performing a sentiment lexicon from the corresponding emoticons.

III. PROPOSED SYSTEM

A. Design

Figure 3.1 shows the design of the proposed system .

The overall process of the framework is as follows:

1. Training set Preparation: First of all we need to prepare a training set containing a pre classified sample tweets. These sample tweets will then undergo a pre-processing stage as follows:

Tokenization: A tweet is generally a sentence containing HTTP symbols, URL, words and emoticons. Tokenization is a process where each tweet is divided into small tokens.

Removal of Http symbols: After tokenizing all the pre classified tweets, the URL and the other HTTP symbols are removed from the tweets. These are removed because these are of least importance as far as classification of tweet is considered.

Removal of Stopwords: Stopword are generally the prepositions or the articles such as ``a'', ``the'', ``of'' etc which does not add any meaning to the tweet and also don't play any major role in classification.

Adding Lexicons: After pre-processing the training set, the sentiment lexicons for the corresponding sentiment words in the pre classified tweets will be added. These lexicons are added from a pre defined domain dictionary. The dictionary contains a sentiment word and its possible variant or synonyms having same meaning. These lexicons are assigned polarity as positive or negative depending on polarity score assigned to them.

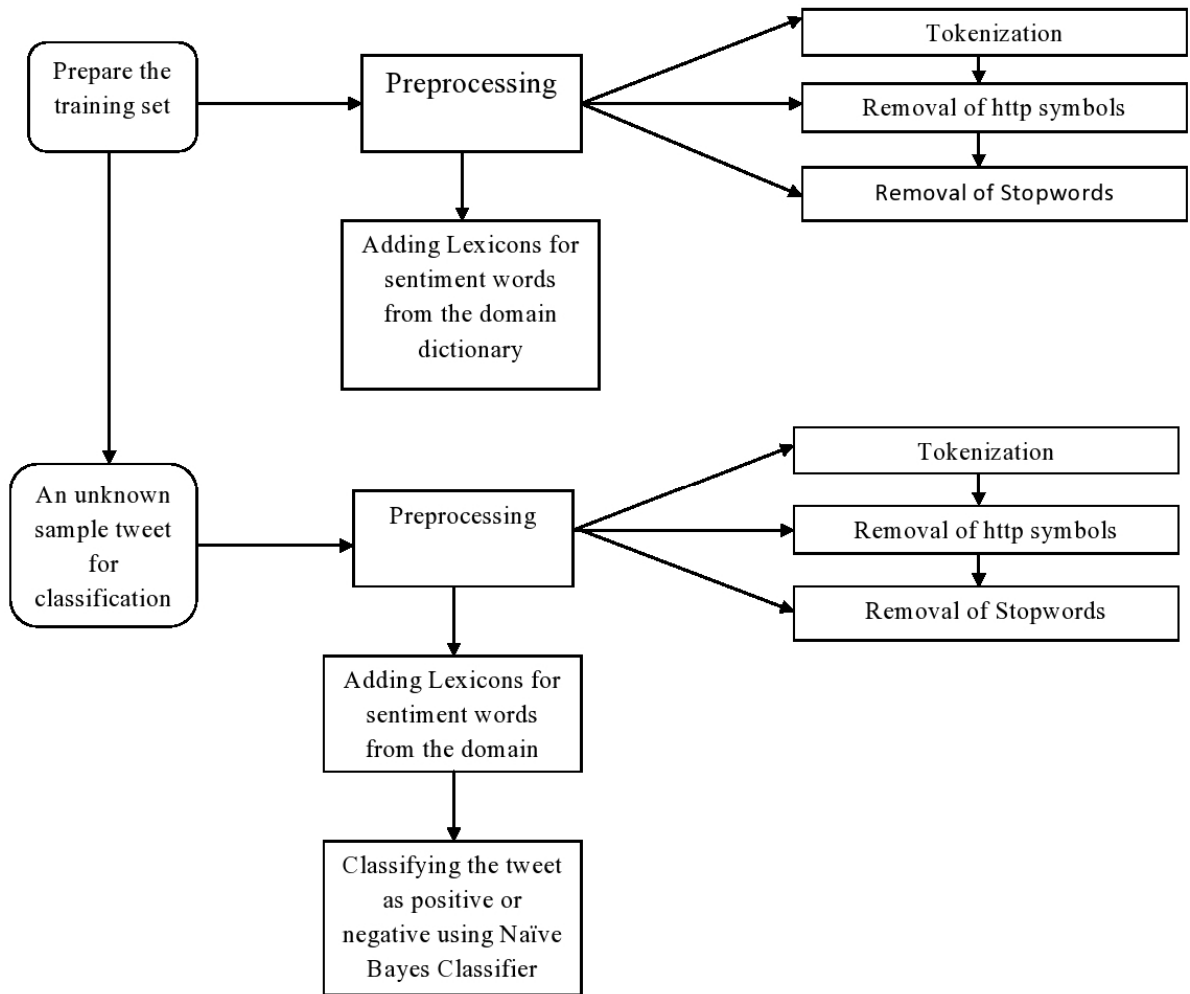


Figure 3.1 Design

Now the training set is prepared for testing. The training set is tested against an unknown sample tweet as follows:

2. Testing an Unknown Sample Tweet

The unknown sample tweet undergoes the same pre-processing and lexicon adding stages as in the case of training set

Tokenization: The sample tweet containing HTTP symbols, words, URL and emoticon is broken down into small tokens.

Removal of HTTP symbols: After tokenization, all the HTTP symbols like the URL and the other HTTP symbols will be removed from the tweet.

Removal of stopwords: After removing the HTTP symbols, stopwords are also removed from the tweet as they are of least importance for classification.

Adding Lexicons: After pre-processing the tweet, a sentiment lexicon is added for each sentiment word present in the tweet. These lexicons are added from the pre defined domain dictionary which

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2015

contains a sentiment word and its corresponding possible synonyms having same meaning. All the lexicons related to the sentiment word present in the tweet are appended at the end of the tweet from the domain dictionary.

Classifying the tweet: The final stage after adding the lexicons is the classification of the tweet. The classification is performed using the NB classifier based on the polarity assign to the lexicon.

B. Multinomial NB Model

Multinomial NB Classifier model is another variant of the traditional NB Model which is explicitly used for document modelling and text classification. The model revolves around two terminologies that form the base of this model. These terminologies are:

a. **Term Frequency (TF):** A term frequency [26] of any given term is defined as the number of times a given term has appear in a particular given text or document. It generally how frequently a term occurs in a document. For example consider a document containing 100 words wherein the word cat appears 3 times. The term frequency (i.e., tf) for cat is then $(3/100) = 0.03$. So formally term frequency can be defined as the number of times a given term t (i.e., word or token) appears in a document d. Also it may happen that since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (the total number of terms in the document) as a way of normalization.

$$\text{Normalized term frequency} = \frac{tf(t,d)}{n_d}$$

where,

tf(t,d) = Term frequency

n_d = The total number of terms in document d.

b. **Term Frequency - Inverse Document Frequency (TF - IDF):** Document frequency of a given term t measures of out all the documents available, how many of them contains the given term t. Inverse document frequency is exactly opposite to document frequency. Inverse Document Frequency measures how important a term is. While computing "TF", all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones. Inverse document frequency is calculated as:

$$\text{Idf}(t) = \log \frac{n_d}{n_d(t)}$$

where,

n_d = The total number of documents.

$n_d(t)$ = The number of documents that contain the term (t).

The term frequency - inverse document frequency (Tf-idf) [26] is used for characterizing text documents. It can be understood as a weighted term frequency, which is especially useful if stop words have not been removed from the text corpus. The Tf-idf approach assumes that the importance of a word is inversely proportional to how often it occurs across all documents. Although Tf-idf is most commonly used to rank documents by relevance in different text mining tasks, such as page ranking by search engines, it can also be applied to text classification via NB. For example, consider a document containing 100 words wherein the word cat appears 3 times. The term frequency (i.e., tf) for cat is then $(3 / 100) = 0.03$. Now, assume we have 10 million documents and the word cat appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as $\log(10,000,000 / 1,000) = 4$. Thus, the Tf-idf weight is the product of these quantities: $0.03 * 4 = 0.12$.

$$\text{Tf - idf} = \text{tf}_n(t,d) * \text{idf}(t)$$

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2015

C. Algorithm

Part A: Training

INPUT: Ts(Preclassified Training set), DI(Lexicon Domain Dictionary), cwords(stopwords)

1. Tokenize the pre classified training set
tz[] = StringTokenizer st = new StringTokenizer(Ts)
2. Remove the url and other special http symbols from the training set
FOR (i=0; i<tz[i].length; i++)
 IF tz[i]. contains url **THEN**
 tz[i].replaceAllTweets(tz[i], "").trim
 ELSEIF tz[i]. contains"@"
 tz[i].replacetweet("@", "")
 END IF
END FOR
3. Removing Stopwords from the training set
FOR (i=0; i<tz[i].length; i++)
 FOR(j=0; j<cwords.length;cwods++)
 IF(tz[i] == cwords[j]) **THEN**
 Replacealltweets(tz[i], "");
 END IF
 END FOR
END FOR
4. Adding Lexicons from the domain dictionary
FOR (i=0; i< tz[i].length;i++)
 FOR (j=0; j< DI[j].length; j++)
 IF tz[i] = DI[j] **THEN**
 Append(tz[i], DI[j])
 ENDIF
 END FOR
END FOR
5. Calculate the normalized *tf* for each term as –
Normalized term frequency = $tf(t; d) / n_d$

Part B: Testing

INPUT: Ts(Sample Test Tweet), DI(Lexicon Domain Dictionary), cwords(stopwords)

6. Select a sample test tweet
7. Tokenize the sample test tweet
sz[] = StringTokenizer st = new StringTokenizer(St)
8. Remove the url and other special http symbols from the sample test tweet
FOR (i=0; i<sz[i].length; i++)
 IF sz[i]. contains url **THEN**
 sz[i].replaceAllTweets(sz[i], "").trim
 ELSEIF (sz[i]. contains"@"
 sz[i].replacetweet("@", "")
 END IF
END FOR
9. Removing Stopwords from the sample test tweet
FOR (i=0; i<sz[i].length; i++)
 FOR(j=0; j<cwords.length;cwods++)
 IF(sz[i] == cwords[j]) **THEN**
 Replacealltweets(sz[i], "");

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2015

- ```
END IF
END FOR
END FOR
10. Adding Lexicons from the domain dictionary
FOR (i=0; i< sz[i].length; i++)
FOR (j=0; j< DI[j].length; j++)
IF sz[i] = DI[j] THEN
Append(sz[i], DI[j])
ENDIF
END FOR
END FOR
```
11. Calculate the normalized *tf* for each term as –  
Normalized term frequency =  $tf(t; d) / n_d$
12. Calculate the positive *idf* and negative *idf* as –  
Inverse Document Frequency =  $\text{Log}(n_d / n_d(t))$
13. Calculate the aggregate *pos\_tf-idf* weight  
 $\text{Pos} = \text{tfn}(t; d) * \text{neg\_idf}(t)$
14. Calculate the aggregate *neg\_tf-idf* weight  
 $\text{Neg} = \text{tfn}(t; d) * \text{neg\_idf}(t)$
15. Calculate the aggregate *tf-idf* weight for entire tweet  
 $\text{Tf-idf} = \text{tfn}(t; d) * \text{idf}(t)$
16. **IF**  $\text{pos} < \text{neg}$  **then**  
classify the tweet as positive  
**ELSE IF**  $\text{neg} < \text{pos}$  **then**  
Classify the tweet as negative  
**ELSE** classify the tweet as neutral

---

#### IV. IMPLEMENTATION DETAILS

**Dataset:** There are not any large public data sets of Twitter messages. The dataset used here for the classification is the standard Sentiment140 dataset. From the Sentiment140 dataset, few tweets have been extracted and are used as a training set to train the classifier. The Sentiment140 dataset is formed using the Application Programming Interface of twitter. The twitter API has parameter that specifies a language. The language parameter is set to English. So our training set contains tweets only in English language form the sentiment140 dataset. To collect positive and negative tweets, the positive and the negative emoticons are used as a query term in the twitter API. For example “:)” and “:-)” both express positive emotion. Similarly, “:(” returns negative tweets. From the large Sentiment140 dataset, tweets related to “Kindle2” which is a amazon tab are extracted and used as a training set to train the classifier.

**Lexicon Domain Dictionary:** Another important factor, from the implementation point of view of the proposed system is the domain dictionary. The dictionary contains two columns, one contains a particular adjective using which a sentiment can be expressed, that is, the sentiment word. The second column contains all the possible synonyms or the forms of words of each particular sentiment word. The classifier is trained to classify the tweet correctly, even if one of the synonyms or corresponding word of a particular sentiment word appears in the tweet.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2015

**Stopwords:** A set of stopwords is formed which also plays important role in the classification of the tweet. Generally, "Stopwords" are the prepositions or the articles in the tweet. For e.g. words such as "of", "not", "is", etc are the words that have the least significance as far as the classification is considered. So a set of stopwords is formed and the sample test tweet is compared with this set of stopwords. All the matched words are eliminated from the tweet, so that the tweet can be optimized to the maximum and the sentiment word can be extracted easily for classification.

## V. EXPERIMENTAL SETUP

The proposed system is implemented using JAVA as a front end technology. At the backend, MySQL 5 is used for database handling. The database is used for storing and retrieving the data related to domain dictionary, term frequencies, positive document frequencies and the negative document frequencies. All the implementation of the proposed system is done using the NetBeans Java IDE, version 8.1.

## VI. EXPERIMENTAL RESULTS

In the proposed system, the classifier is first trained using the training set and then it is tested using sample test tweet. To verify, whether the lexicon based approach shows better results than the existing one, an experiment is carried out. In the experiment, a set of twenty tweets tweeted by the users in three different months is collected. This set of total sixty tweets contain tweets related to kindle2. After collecting the tweets, the tweets are tested for classification. The classification is done using both the approaches, that is , the existing approach and the proposed Lexicon based approach with NB classifier.

Table 4.1 shows the comparative analysis of the experimental results. Out of the 60 tweets collected from three months, the proposed lexicon based approach shows an average improvement of 10% over the conventional existing approach

Table 4.1 Comparative Analysis

| Month | Total number of tweets | Number of tweets classified accurately using existing approach | Number of tweets classified accurately using lexicon approach |
|-------|------------------------|----------------------------------------------------------------|---------------------------------------------------------------|
| June  | 20                     | 10                                                             | 12                                                            |
| Sept  | 20                     | 10                                                             | 14                                                            |
| Dec   | 20                     | 11                                                             | 15                                                            |

Tweets regarding the amazon's new app "Kindle2" were collected. Total 60 tweets of three months were collected, 20 tweets of each month. These 60 tweets were then tested with the existing approach of Naive bayes classifier as well as the proposed combinational approach of NB classifier with Lexicon based approach. The proposed approach therefore shows improvement by 10%.



# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2015

Figure 4.1 shows the graphical comparative analysis of the tweet classification. From the graph it is very clear that the proposed approach classifies more tweets accurately than the conventional existing approach

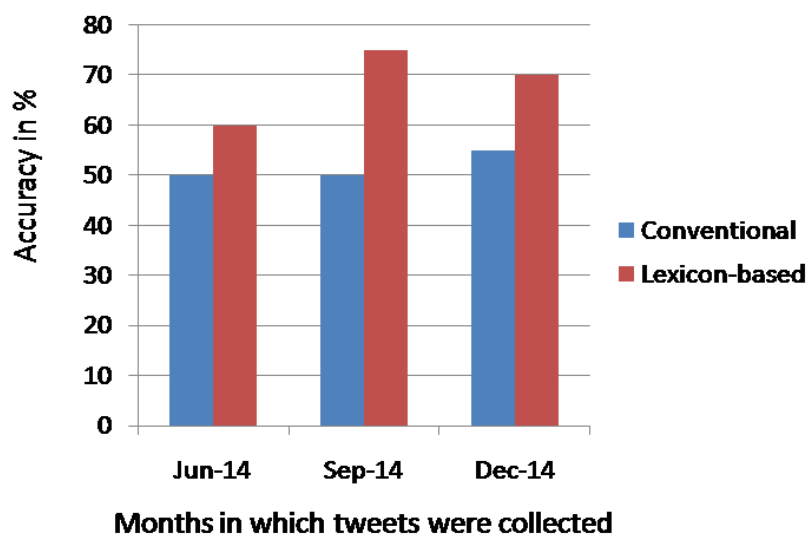


Figure 4.1 Improved accuracy of the classifier

## VII. CONCLUSION

NB classifier with lexicon based approach is implemented in the proposed system instead of the conventional NB classifier alone. Existing approach had two main problems. First, the classifier cannot classify the tweet accurately, if the sentiment words used in the sample test tweet and that in the training set are not same. Even if the sentiment word used in the sample test tweet has the same meaning as that of the word in the training set, the tweet is not classified correctly. The domain dictionary used in the lexicon based approach solves this problem. The domain dictionary contains some of the most frequently used sentiment words and their possible synonyms or the variants having same meaning. Adding lexicons to the tweet adds the corresponding synonyms of the particular sentiment word in the test tweet and classifies accurately and improve the accuracy of the classifier.

The second problem is the flexibility of using different variety of sentiment words. The lexicon based approach also solves the second problem as it allows user to use variety of sentiment words due to its concept of domain dictionary. The proposed approach has the ability to increase the accuracy of the classifier and provide flexibility to the user in giving a tweet with variety of sentiment words.

## REFERENCES

- [1] Raymond Kosala, Hendrik Blockee, "Web Mining Research: A Survey", ACM, Vol.1, pp.1-15, 2000.
- [2] Sandra Stendahl, Andreas Andersson, Gustav Strömberg, "Web Mining", pp. 1-7.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2015

- [3] D. Jayalatchumy, Dr. P.Thambidurai, "Web Mining Research Issues and Future Directions – A Survey", IOSR Journal of Computer Engineering (IOSR-JCE), Vol.14, pp. 20-27, 2013.
- [4] Abdelhakim Herrouz, Chabane Khentout, Mahieddine Djoudi, "Overview of Web Content Mining Tools", The International Journal of Engineering And Science, Vol 2, pp 1-6, 2013.
- [5] R.Malarvizhi, K.Saraswathi, "Web Content Mining Techniques Tools and Algorithms – A Comprehensive Study", International Journal of Computer Trends and Technology, Vol4, pp. 2940-2945, 2013.
- [6] Arvind Arasu, Hector Garcia-Molina, "Extracting Structured Data from Web Pages", ACM, pp. 337-348, 2003
- [7] Michael J. Cafarella, Jayant Madhavan, Alon Halevy, "Web-Scale Extraction of Structured Data" ACM, vol. 37, Issue 4, pp. 55-61, 2008.
- [8] Nilesh Dalvi, Ashwin Machanavajjhala, Bo Pang, "An Analysis of Structured Data on the Web", International Conference on Very Large Data Bases, pp. 680-691, 2012.
- [9] Miloš Radovanovic, Mirjana Ivanovic, "TEXT MINING: APPROACHES AND APPLICATIONS", Novi Sad J. Math. pp. 227-234, 2008.
- [10] Ian H. Witten, "Text Mining" pp. 1-23.
- [11] Vishal Gupta, Gurpreet S. Lehal, "A Survey of Text Mining Techniques and Applications", JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, Vol.1, pp.60-76, 2009.
- [12] Xiaowen Ding, Bing Liu, Philip S. Yu, "A Holistic Lexicon-Based Approach to Opinion Mining", ACM, 2008.
- [13] Ion SMEUREANU, Cristian BUCUR, "Applying Supervised Opinion Mining Techniques on Online User Reviews", Informatica Economic, Vol 16, 2012.
- [14] K.NATHIYA, Dr.N.K.Sakthivel, "Development of an Enhanced Efficient Parallel Opinion Mining for Predicting the Performance of Various Products", International Journal of Innovative Research in Computer and Communication Engineering, Vol.1, 2013.
- [15] Arti Buche, Dr. M. B. Chandak, Akshay Zadgaonkar, "OPINION MINING AND ANALYSIS: A SURVEY", International Journal on Natural Language Computing, Vol.2, 2013.
- [16] Poongodi S, Radha N, "Classification of user Opinions from tweets using Machine Learning Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Vol.5, 2013.
- [17] Amandeep Kaur, Vishal Gupta, "A Survey on Sentiment Analysis and Opinion Mining Techniques", JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, Vol.5, 2013.
- [18] Ayesha Rashid, Naveed Anwer, Dr. Muddaser Iqbal, Dr. Muhammad Sher, "A Survey Paper: Areas, Techniques and Challenges of Opinion Mining", International Journal of Computer Science Issues, Vol.10, 2013.
- [19] Pravesh Kumar Singh, "ANALYTICAL STUDY OF FEATURE EXTRACTION TECHNIQUES IN OPINION MINING", International Journal on Soft Computing, 2013.
- [20] I.Hemalatha, Dr. G. P Saradhi Varma, Dr. A.Govardhan, "Sentiment Analysis Tool using Machine Learning Algorithm", International Journal of Emerging Trends and Technology in Computer Science, Vo.:2, 2013.
- [21] Ana C.E.S.Lima, Leandro N. De Castro, "Automatic Sentiment analysis of Twitter Messages", 2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN), IEEE, pp. 52-57, 2012.
- [22] S.L. Ting, W.H. Ip, Albert H.C. Tsang, "Is Naïve Bayes a Good Classifier for Document Classification", International Journal of Software Engineering and Its Applications, Vol.5, pp.37-46, 2011.
- [23] Hiroshi Shimodaira, "Text Classification using Naïve Bayes", 2015.
- [24] Alexander Hogenboom, Daniella Bal, Flavius Frasinca, Malissa Bal, Franciska de Jong, Uzay Kaymak, "Exploiting Emoticons in Sentiment Analysis", 2013.
- [25] Bing Liu, "Sentiment Analysis and Opinion Mining", Morgan and Clypool Publishers, 2012.