Performance Analysis of Sentimental Data Using Machine Learning Algorithms

M.Vadivukarassi[1] N.Puviarasan[2], P.Aruna[3]
[1 ,3] *(Department of Computer Science and Engineering, Annamalai University, India)*
[2] *(Department of Computer and Information Science, Annamalai University, India)*

Abstract: *Social media allow creation, sharing and exchange of tweets among people, business and products. The main goal is to determine the classifiers and features produce the best results for this particular sentiment classification task. The tweets are classified 'positive', 'negative' and 'neutral' experimentally with two supervised classifiers such as Naïve Bayes and Random forest. The experiments are executed in two stages: the first stage uses Bag of words (BOW) feature set, and the second stage uses extended feature (EF) sets respectively with different classifiers. These results demonstrate that whole sentiment analysis process, providing high accuracy levels in Random forest classifier with the extended feature extraction method.*

Keywords: *Bags of words, Extended features, Naïve Bayes, Polarity, Random Forest, Sentimental analysis, Twitter.*

I. Introduction

Social media have become a way of expressing a collective opinion and interest of the user. Among the many social media websites, Twitter is a trendy micro blog service, designed for simplified communication. There are about 250 billion tweets posted daily on Twitter, approximately 100, 000 tweets per minute, illustrating this service as an essential repository for data analysis. This is the useful source of content for psychologists, marketers and others interested in the extraction and mining of opinions, views, attitudes and moods [1]. Many investigations are being performed around Twitter, such as sentiment analysis, event detection, text classification and others. The sentiment analysis provides a view about the sentiment expressed in the tweets. This sentiment is basically labeled according to the tweet polarity, that is, whether the text has a positive, negative or neutral suggestion. This measure helps to observe the public and market opinion about itself for industries. This task is named as polarity analysis or sentiment polarity [2]. The main objective of this research paper is to predict sentiment for the given tweets using python script. The paper is organized as follows. Section 2 discussed on review of the related works on the sentiment analysis, and Section 3 describes the proposed framework and Section 4 presents and discusses the obtained results. The paper concludes in Section 5 with a general conclusion about the proposal and avenues for future investigation.

II. Related Works

They surveyed mainly on polarity analysis of twitter data which is used to analyze the tweets where interest are highly structured, heterogeneous and are either positive or negative, or neutral. They mainly discussed on the various types of machine learning algorithms like Naive Bayes, Max entropy that may not produce accurate results for either of unigrams, bigrams or weighted unigrams [3]. The authors aimed to review some papers regarding research in sentiment analysis on Twitter, describing the methodologies adopted and models applied, along with describing a generalized Python based approach. It focuses on analyzing the sentiments of the tweets and feeding the data to a machine learning model in order to train it and then check its accuracy, so that we can use this model for future use according to the results [4]. The unigrams and bigrams and apply Term Frequency Inverse Document Frequency (TF-IDF) to find the weight of a particular feature in a text is used and hence filter the features having the maximum weight. The TF-IDF is a very efficient approach and is widely used in text classification and data mining [5]. Classifying the tweets according to the sentiment are focused and expressed using a supervised classification approach. The classifier used is Naïve Bayes Classifier to classify the tweets as positive, negative or neutral. The classifier is trained using tweets which bear a distinctive polarity. The percentage of the positive and negative tweets are then computed and is represented graphically. The result can be used further to gain an insight into the views of the people using twitter about a particular topic that is being searched by the user [6]. Many applications which use Random Forest to classify the dataset like Network intrusion detection, Email spam detection, gene classification, Credit card fraud detection, and Text classification are discussed. The real implementation of the Random Forest algorithm with the stepwise procedure is explained and also the results are discussed. Actual Random Forest Algorithm and its features are also discussed to highlight the main features of Random Forest Algorithm more clearly [7].

Improved-RFC (Random Forest Classifier) approaches for multi-class disease classification problem is presented. It consists of a combination of Random Forest machine learning algorithm, an attribute evaluator method and an instance filter method. It intends to improve the performance of the Random Forest algorithm. The performance results confirm that the proposed improved-RFC approach performs better than Random Forest algorithm with increase in disease classification accuracy up to 97.80% for multi-class groundnut disease dataset [8].

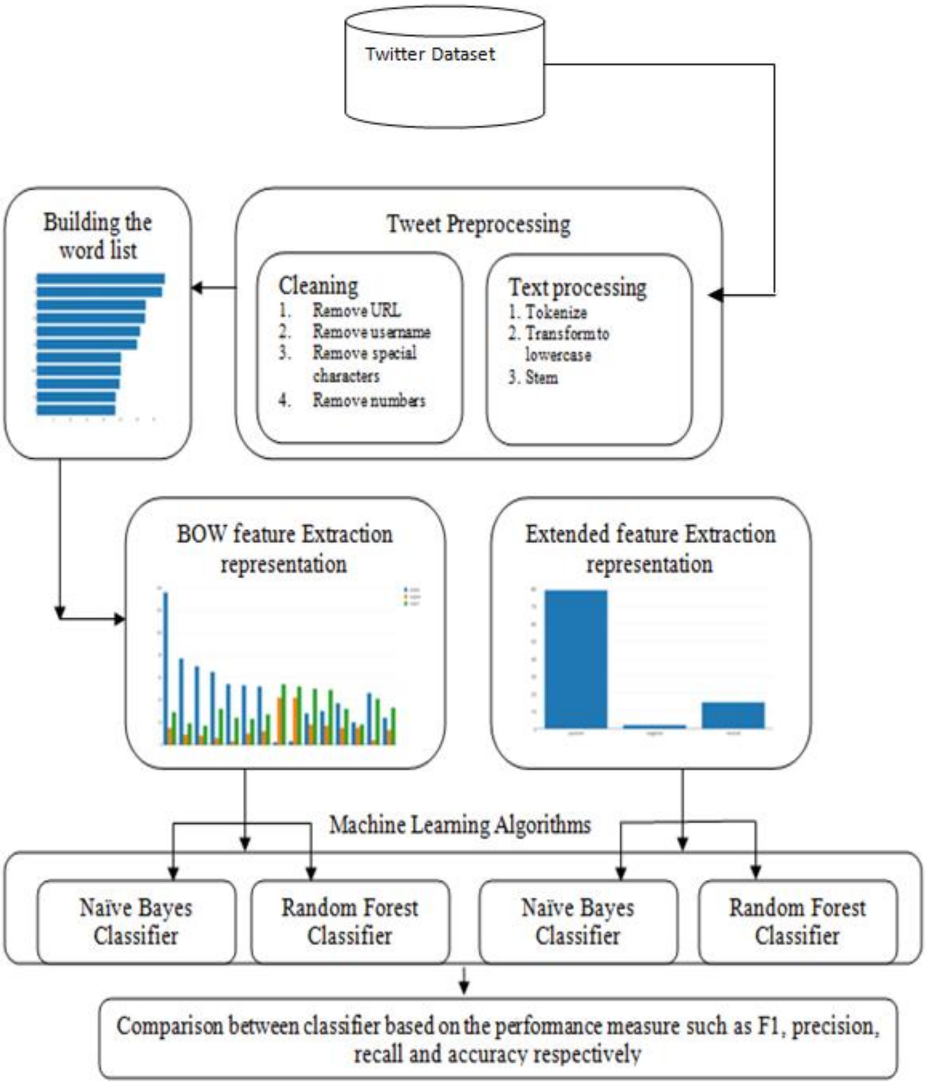

Twitter Dataset
Building the word list
Tweet Preprocessing
Cleaning
1. Remove URL
2. Remove username
3. Remove special characters
4. Remove numbers
Text processing
1. Tokenize
2. Transform to lowercase
3. Stem
BOW feature Extraction representation
Extended feature Extraction representation
Machine Learning Algorithms
Naïve Bayes Classifier
Random Forest Classifier
Naïve Bayes Classifier
Random Forest Classifier
Comparison between classifier based on the performance measure such as F1, precision, recall and accuracy respectively

Fig.1. Block diagram of the proposed system

III. Proposed System

This section describes the overall structure for capturing and analyzing tweets streamed in real time to achieve sentiment analysis with different classifier. Fig. 1 shows the block diagram of the proposed system. It comprises stages such as data collection, preprocessing, feature extraction and classification. A dataset is created using twitter posts of mobile reviews. These tweets are preprocessed and feature vector is created using relevant features. Finally, using different classifiers, tweets are classified into positive, negative and neutral classes. In this paper, the real-time data are collected from Twitter with the help of Twitter streaming API based on particular area based on mobile reviews and save in the Twitter dataset.

Data Collection of tweets

Twitter contains two APIs – Twitter Search API and Twitter Streaming API. The Search API allows users to query against the indices of popular tweets and captures the only the past week of relevant tweets and the rate limits of tweets are 180 requests query per 15 minutes. In contrast to the Search API, the Streaming API can provide the user for getting the real-time stream of tweets [9]. In this paper, the tweets are collected in the particular place using Twitter streaming API with the help of Tweepy libraries in Python. The collected tweets contain the attributes such as Tweet_ID, Username, Created_at, Text, Time and Geo_cordinates respectively. The tweets are only extracted from this database and further used to process such as tweet preprocessing and feature extraction methods.

Data Preprocessing of tweets

The collected real-time data from Twitter are used for preprocessing method before extracting the features. Typically the tweets consists of message along with usernames, empty spaces, special characters, stop words, emotions, abbreviations, hash tags, time stamps, URL's, etc. These tweets are preprocessed by using NLTK techniques. In these techniques, the tweets are first extracted and then all unwanted contents are removed. The pre-processing is done in such a way that data represented only in terms of words that can easily classify the tweets. A Python code is created in which a function is defined will be used to get processed tweeted.

Feature Extraction of tweets

Feature extraction can be used to extract feature vectors from tweets. In this paper, different features such as Bags of Words (BOW) and proposed extended features are used for extracting the features. A python script is created to extract the features of the training data.

Bag of Words (BOW) Representation

It is the process which ignores the order of the words and generates a reduced form of the tweets containing the number of occurrences or the frequency of each word in the tweets. This method consists of count vector and Time Frequency and Inverse Document Frequency (TF-IDF). In this paper, TF-IDF is used. This value increases proportionally to the number of times a word appears in a data set which helps to adjust that some words appear more frequently in general.

$$TF = \frac{Number\ \ of\ times\ that\ word\ appears}{Number\ \ of\ words\ in\ the\ tweet} \quad (1)$$

$$IDF = \frac{Number\ \ of\ tweets}{Number\ \ e\ tof\ words\ in\ the\ tweet} \quad (2)$$

$$TF - IDF = TF * IDF \quad (3)$$

i. Extended features (ET) representation

In this paper, the special characters like exclamation marks, question mark, upper case, etc. are extracted from the tweets for determining the sentiment. The extraction of those features must be done before the preprocessing steps. For this purpose of emoticons, the EmoticonDetector class is created. The file emotions. txt contains a list of positive and negative emotions, which are used in the tweets. The feature name and explanation of the features are described in the Table 1.

Tweet Classification

Machine Learning techniques use a training set and a test set of classification. To classify the tweets in different classes, the different classifier has been built in the proposed research work. To build the classifier a library of Python called, Scikit-learn is used. Scikit-learn are very powerful and most useful library in Python which provides many classification algorithms. Scikit-learn also include tools for classification, clustering, regression and visualization. In this paper, the machine learning algorithms such as Bernoulli Naïve-Bayes and Random Forest are used to classify the tweets as positive, negative and neutral.

Table 1: List of extended features with explanations

Feature name	**Explanation**
Number of upper case	People tend to express with either positive or negative emotions by using a lot of uppercase words
Number of !	Exclamation marks are likely to increase the strength of opinion
Number of ?	Might distinguish neutral tweets - seeking for information
Number of quotations	Same as above
Number of mentions	Sometimes people put a lot of mentions on positive tweets, to share something good
Number of hashtags	Just for the experiment
Number of urls	Similar to the number of mentions

Bernoulli Naïve Bayes classifier

Bernoulli Naïve Bayes implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features, but each one is assumed to be a binary-valued (Bernoulli, boolean) variable. Therefore, this class requires samples to be represented as binary-valued feature vectors; if handed any other kind of data, a Bernoulli Naïve Bayes instance may binarize its input (depending on the binary parameter).

$$P\left(\frac{x_i}{y}\right) = P\left(\frac{i}{y}\right)x_i + \left(1 - P\left(\frac{i}{y}\right)\right)(1 - x_i) \qquad (4)$$

The decision rule for Bernoulli naive Bayes is based on which differs from multinomial NB's rule in that it explicitly penalizes the non-occurrence of a feature i that is an indicator for class y, where the multinomial variant would simply ignore a non-occurring feature. In the case of tweet classification, word occurrence vectors may be used to train and use this classifier [10].

Random Forest Classifier

Random forest can be used for both classification and regression kind of problems. It is a supervised classification algorithm and creates the forest with a number of trees. In random forest classifier, the higher the number of trees in the forest gives higher accuracy results. The steps in the random forest algorithm can split into two stages such as creation and prediction. With the creation of the random forest algorithm, it starts with randomly selecting "k" features out of total "m" features. The observations are observed and randomly selected "k" features are used to find the root node by using the best split approach. Then the daughter nodes are also calculated using the same approach. The first 3 stages are forms the tree with a root node and having the target as the leaf node. Finally, the step 1 to 4 stages are repeated to create "n" randomly created trees. This randomly created tree creates the random forest.The steps for the creation and prediction of Random Forest are as follows:

Creation

1. Randomly select "k" features from total "m" features where k << m
2. Among the "k" features, calculate the node "d" using the best split point.
3. Split the node into daughter nodes using the best split.
4. Repeat 1 to 3 steps until "l" number of nodes has been reached.
5. Build forest by repeating steps 1 to 4 for "n" number of times to create "n" number of trees.

Prediction

1. Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target).
2. Calculate the votes for each predicted target.
3. Consider the high voted predicted target as the final prediction from the random forest algorithm.

valuation metrics widely used to evaluate the effectiveness of classification algorithms on a given category. The Precision (P), is the number of correctly classified positive tweets divided by the number of tweets labeled as positive by the system. It defines as:

$$P = \frac{True\ Positive}{True\ Positive + False\ Positive} \qquad (5)$$

The Recall (R), is the number of correctly classified positive tweets divided by the number of positive tweets in the dataset. It defines as:

$$R = \frac{True\ Positive}{True\ Positive + False\ Negative} \qquad (6)$$

Given P and R, the F-measure is defined as:

$$F = 2.\frac{P*R}{P+R} \qquad (7)$$

To ensure reliable results, all of the experiments were conducted using a ten-fold cross validation method [12].

IV. Results And Discussion

The tweets are collected using Twitter Streaming API and saved in the database. The input data consisted two CSV files: train.csv (1000 tweets) and test.csv (700 tweets) - one for training and one for testing. All tweets are in English collected from the country India about movie reviews in the month of June, July and August 2017 respectively. It simplifies the processing and analysis of the tweets and also the distinction between testing and training data. The training set had the following distribution as in Fig 2. This figure visualized that number of positive, negative and neutral reviews are 500, 150 and 350 respectively. The tweet attribute such as ID, emotion, and text is only extracted separately as shown in Fig 3. These attributes are loaded and used for sentimental distribution analysis. The preprocessing of the tweet such as cleaning, text processing is used and the top word list is created as shown in the following Fig 4, Fig 5 and Fig 6 respectively. The word list (dictionary) is built by a simple count of occurrences of every unique word across all of the training data set. Fig 5 represents the tokenizing and steaming method of the tweets.

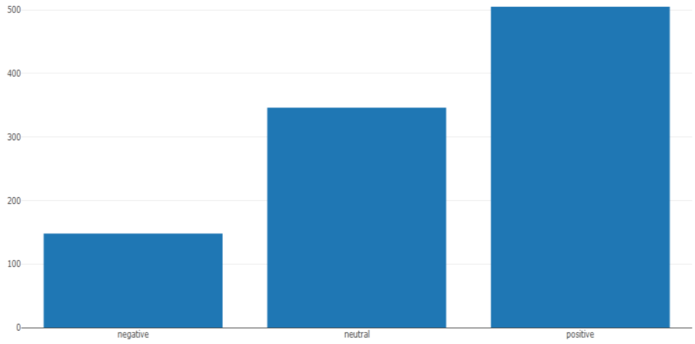
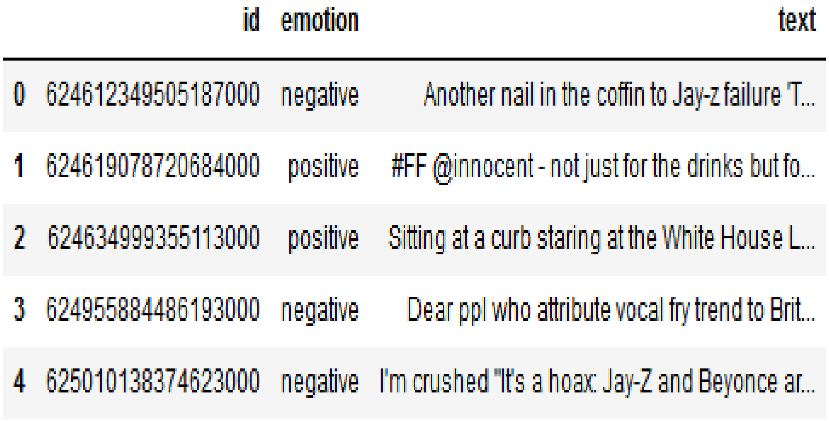
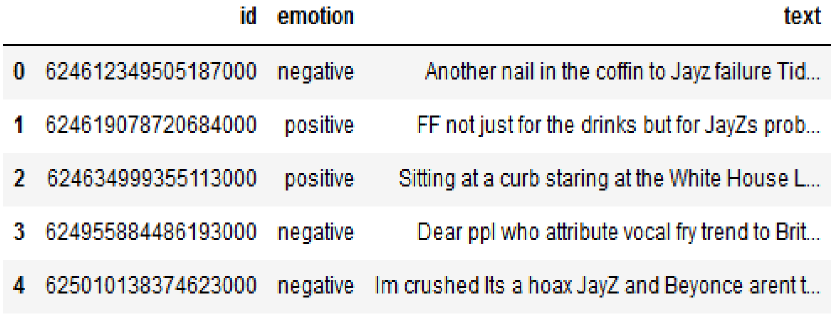


Fig. 2. Sentiment type distribution in the training sets



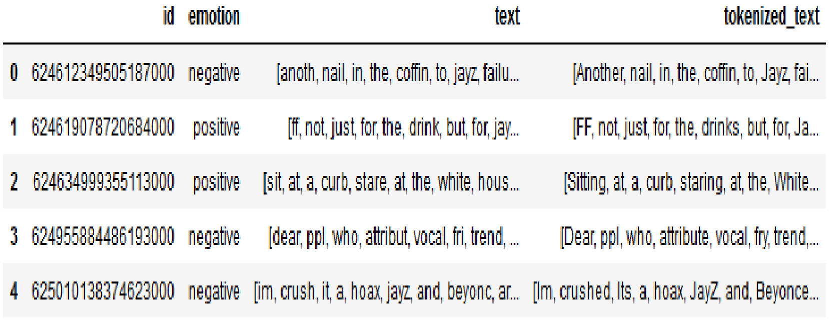
	id	emotion	text
0	624612349505187000	negative	Another nail in the coffin to Jay-z failure 'T...
1	624619078720684000	positive	#FF @innocent - not just for the drinks but fo...
2	624634999355113000	positive	Sitting at a curb staring at the White House L...
3	624955884486193000	negative	Dear ppl who attribute vocal fry trend to Brit...
4	625010138374623000	negative	I'm crushed "It's a hoax: Jay-Z and Beyonce ar...

Fig.3. Loading of the tweets



	id	emotion	text
0	624612349505187000	negative	Another nail in the coffin to Jayz failure Tid...
1	624619078720684000	positive	FF not just for the drinks but for JayZs prob...
2	624634999355113000	positive	Sitting at a curb staring at the White House L...
3	624955884486193000	negative	Dear ppl who attribute vocal fry trend to Brit...
4	625010138374623000	negative	Im crushed Its a hoax JayZ and Beyonce arent t...

Fig.4. Cleaning of the tweets



	id	emotion	text	tokenized_text
0	624612349505187000	negative	[anoth, nail, in, the, coffin, to, jayz, failu...	[Another, nail, in, the, coffin, to, Jayz, fai...
1	624619078720684000	positive	[ff, not, just, for, the, drink, but, for, jay...	[FF, not, just, for, the, drinks, but, for, Ja...
2	624634999355113000	positive	[sit, at, a, curb, stare, at, the, white, hous...	[Sitting, at, a, curb, staring, at, the, White...
3	624955884486193000	negative	[dear, ppl, who, attribut, vocal, fri, trend, ...	[Dear, ppl, who, attribute, vocal, fry, trend,...
4	625010138374623000	negative	[im, crush, it, a, hoax, jayz, and, beyonc, ar...	[Im, crushed, Its, a, hoax, JayZ, and, Beyonce...

Fig 5: Tokenization & stemming of the tweets

Fig 6 represents the top wordlist of the all tweets along with the number of occurrences. It shows that the word ‘not’ had occurred 380 times and the word ‘day’ had occurred 370 times respectively in the twitter data.

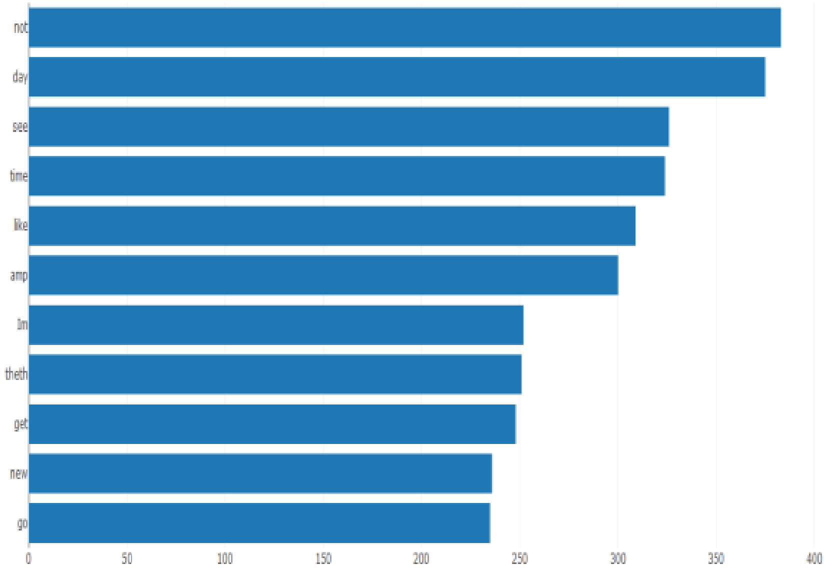
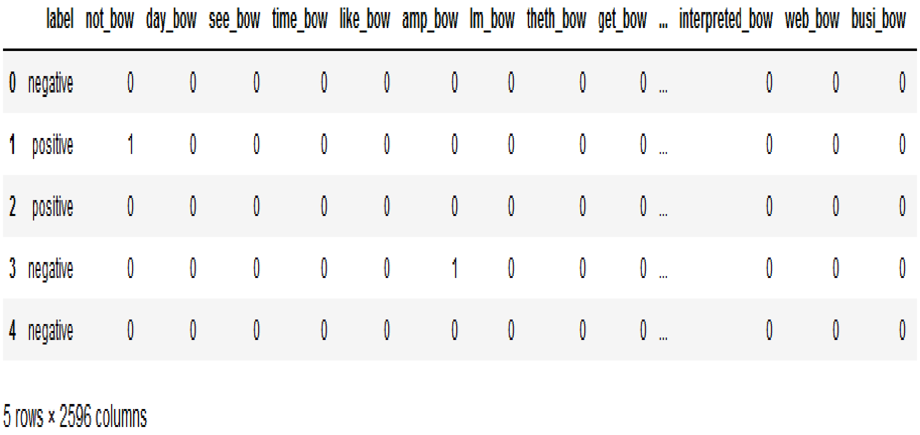


Fig 6: Building the top wordlist

Fig 7 shows the words that are more common for particular sentiments and visualizes the top wordlist present in the each tweet using BOW feature extraction method. It represents the common feature vector present in the each tweet.

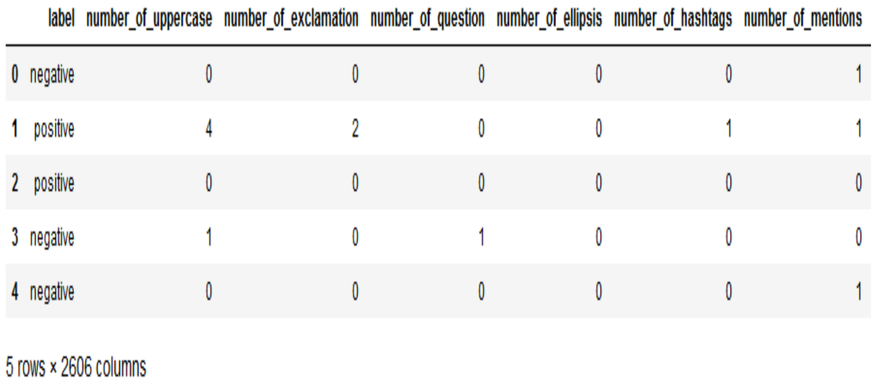


	label	not_bow	day_bow	see_bow	time_bow	like_bow	amp_bow	Im_bow	theth_bow	get_bow	...	interpreted_bow	web_bow	busi_bow
0	negative	0	0	0	0	0	0	0	0	0	...	0	0	0
1	positive	1	0	0	0	0	0	0	0	0	...	0	0	0
2	positive	0	0	0	0	0	0	0	0	0	...	0	0	0
3	negative	0	0	0	0	0	1	0	0	0	...	0	0	0
4	negative	0	0	0	0	0	0	0	0	0	...	0	0	0

5 rows × 2596 columns

Fig 7: Common words for particular sentiments

Fig 8 shows the sample of common extra features for particular tweets using the ET feature extraction method. These extended features are separate the dataset such as number exclamation marks, number of upper cases, etc. Those features occur only in a small subset of the training dataset respectively.



	label	number_of_uppercase	number_of_exclamation	number_of_question	number_of_ellipsis	number_of_hashtags	number_of_mentions
0	negative	0	0	0	0	0	1
1	positive	4	2	0	0	1	1
2	positive	0	0	0	0	0	0
3	negative	1	0	1	0	0	0
4	negative	0	0	0	0	0	1

5 rows × 2606 columns

Fig.8. Common extra features for particular sentiments

Fig 9 represents the graphical visualization of the most common words across sentiments for positive, negative and neutral. Here, the toppest word 'Jurass' is used by the twitter user in positive, negative and neutral tweets. It shows that Jurassic Park is the movie posted by the twitter user in real time at the particular place respectively.

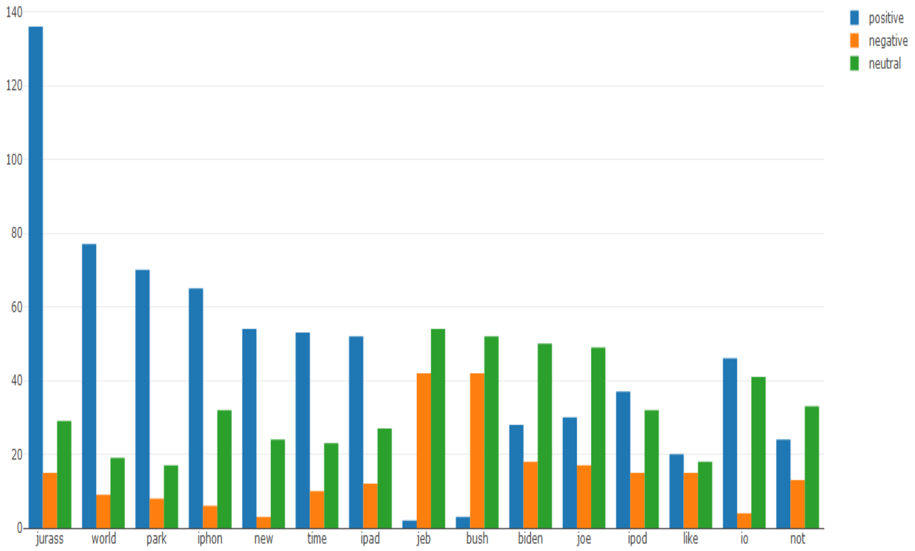


Fig.9. Visualization of most common words across sentiments

In the experimental analysis BOW and EF features are added with the machine learning classifier such as Bernoulli Naïve Bayes and Random Forest separately and the performance measures such as F1, precision, recall and accuracy scores are calculated. Table 2 shows the learning and predicting the time of classifiers with different feature extraction method. It represents that the learning and predicting the time of Bernoulli Naïve Bayes classifier with BOW features is greater than the EF feature extraction method.

Table 2. Learning and predicting the time of classifiers

Classifier	Feature Extraction	Learning Time(sec)	Predicting Time(sec)
Bernoulli Naïve Bayes	BOW	0.0650	0.0220
	EF	0.0610	0.0200
Random Forest	BOW	2.1381	0.3120
	EF	1.9741	0.2060

Similarly the learning and predicting the time of Random Forest classifier with BOW features is greater than the EF feature extraction method. From this Table 2, it shows that classifier with ET feature extraction method produces good results compared to the BOW feature extraction method.

Table 3. Performance measure sof classifier based on Precision, recall, F-measure

Classifier	**Feature Extraction**	**Polarity**	**F1**	**Precision**	**Recall**	**Average (%)**
Bernoulli Naïve Bayes	BOW	Positive	0.7240	0.6009	0.9104	57.40
		Negative	0	0	0	
		Neutral	0.4074	0.4925	0.3473	
	EF	Positive	0.7267	0.6080	0.9029	57.77
		Negative	0	0	0	
		Neutral	0.4216	0.4929	0.3684	
Random Forest	BOW	Positive	0.6896	0.6410	0.7462	56.29
		Negative	0.0444	0.2500	0.0243	
		Neutral	0.4975	0.4636	0.5368	
	EF	Positive	0.7181	0.6524	0.7985	60
		Negative	0.0950	1	0.0487	
		Neutral	0.5326	0.5096	0.5578	

From Table 3 it clearly shows that Random Forest method shows higher classification accuracy when compared to Bernoulli Naïve Bayes. On account, Random forest with ET feature extraction method gives good results on a dataset of small sample size. Fig 10 clearly explains the obtained F1 score, precision and recall obtained by these classifiers based on sentimental analysis. Thus, this experimental analysis proved that the RF with BOW features is lesser than the RF with EF features such as 56.29% and 60% respectively.

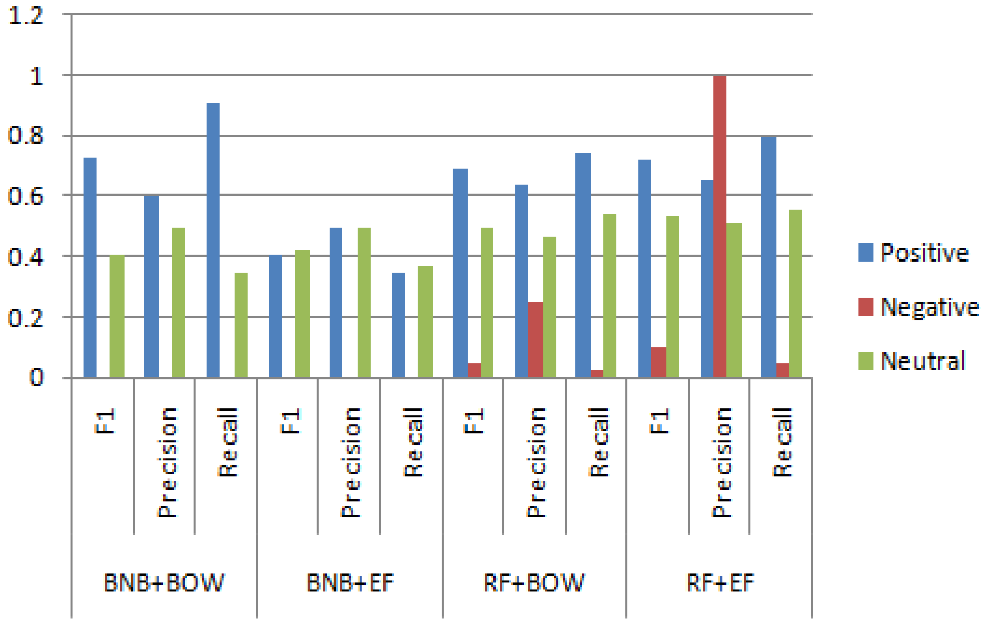


Fig.10. Graphical analysis of performance measures

Table 4. Average accuracy of classifiers after cross validation process

Classifier	**Feature Extraction**	**Cross validation Completed(sec)**	**Average Accuracy (%)**
Bernoulli Naïve Bayes	BOW	7.4854	57
	EF	20.4561	56
Random Forest	BOW	6.1060	58
	EF	19.0710	60

Table 4 shows the average accuracy of classifiers after cross validation process. Here, the performance of BNB classifiers with BOW features produces the accuracy at 57.40%. In order to optimize speed of testing on machine, 8-fold cross validation is used. This cross-validation combines (averages) measures of fit (prediction error) to derive a more accurate estimate of model prediction performance. It shows 56.86% of average accuracy respectively. Similarly the performance of BNB classifiers with ET features produces the accuracy at 57.77 % and average accuracy after cross validation as 53.65% which is lesser respectively. Thus, these experimental results proved that after cross validation method, RF with EF features produces best average accuracy compared to BOW features such as 57% and 56.42% respectively. This analysis, performed that the Bernoulli Naïve Bayes classifier is better compared to the Random Forest classifier. To get Random Forest classifier as better accuracy, the extended feature(ET) representation is used. Fig 11 describes that the Naïve Bayes classifier with BOW features are less compared to the Random Forest classifier with extended features. In this paper, the experiment proved that extended features are easy for the classifiers compared to the Bag of words feature representation method.

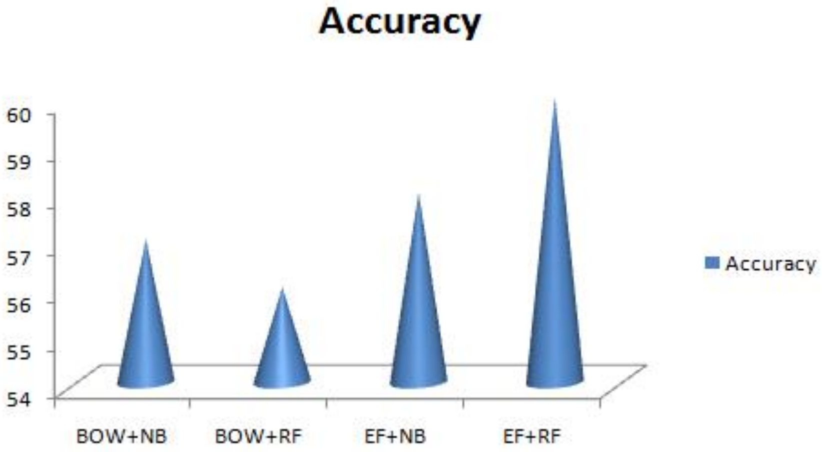


Fig.11. Comparison of accuracy using classifiers

V. Conclusion

Twitter is the largest source of data, which make it more attractive for performing sentiment analysis. It mainly focuses on analyzing the sentiments of the tweets and feeding the data to a machine learning model in order to train it and then check its accuracy. It comprised of steps like data collection, text pre-processing, sentiment detection, sentiment classification, training and testing the model. The twitter data are collected and performs preprocessing using Natural Language Processing based techniques, followed by a feature extraction method in order to extract sentiment related features. Finally, a model is trained using machine learning classifiers like Bernoulli Naïve Bayes, Random Forest and tests on test data. The performance of the model can be calculated in terms of accuracy, precision, recall and F1-score. The proposed framework performs sentiment analysis using Bernoulli Naive Bayes and Random Forest algorithms. The results show that Random Forest performs extremely well, showing high accuracy for classifying sentiment of tweets. The proposed tweet analytic framework is also real-time, fast, scalable, and reliable.

Acknowledgements

This work was supported by Department of Computer Science and Enginerring Annamalai University. The authors Dr.N. Puviarasan and Dr. P.Aruna had helped a lot to do additional experiments for revising the manuscript and I would like to express warm thanks to them.

References

[1]. S.Bhuta, A.Doshi, U.Doshi, M.Narvekar, A review of techniques for sentiment analysis of Twitter data , in: Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on,Ghaziabad,India,2014,pp.583–591

[2]. Pranav Waykar, Kailash Wadhwani, Pooja More," Sentiment analysis in twitter using Natural Language Processing (NLP) and classification algorithm", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 5 Issue 1, January 2016.

[3]. V.Lakshmi, K.Harika , H.Bavishya, Ch.Sri Harsha ," Sentiment Analysis Of Twitter Data" , International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 02 , Feb -2017.

[4]. Bhumika Gupta, Monika Negi, Kanika Vishwakarma, Goldi Rawat, Priyanka Badhani," Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python", International Journal of Computer Applications (0975 – 8887), Volume 165 – No.9, May 2017.

[5]. Peiman Barnaghi, John G. Breslin and Parsa Ghaffari, "Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment", IEEE Second International Conference on Big Data Computing Service and Applications, 2016.

[6]. Pranav Waykar, Kailash Wadhwani, Pooja More," Sentiment analysis in twitter using Natural Language Processing (NLP) and classification algorithm", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 5 Issue 1, January 2016.

[7]. Mohammed Zakariah," Classification of large datasets using Random Forest Algorithm in various applications: Survey", International Journal of Engineering and Innovative Technology (IJEIT) Volume 4, Issue 3, September 2014.

[8]. Archana Chaudhary , Savita Kolhe , Raj Kamal," An improved random forest classifier for multi-class classification", Information Processing In Agriculture 3 (2016) 215–222.

[9]. M.Vadivukarassi, Dr.P.Aruna, N.Puviarasan,"Real Time Visualization Of tweets from Twitter Stream Analysis using Python", International Journal of Applied Engineering Research(IJAER), ISSN 0973-4562,Vol. 11, No.4, pp: 490-500, 2016.(SCOPUS Indexed Journal) .

[10]. Daniel Jurafsky & James H. Martin," Naive Bayes and SentimentClassification ", Speech and Language Processing., August 7, 2017.

[11]. Archana Chaudhary , Savita Kolhe , Raj Kamal," An improved random forest classifier for multi-class classification", Information Processing In Agriculture 3 (2016) 215–222.

[12]. M. Vadivukarassi, N. Puviarasan and P. Aruna*,"* Sentimental Analysis of Tweets Using Naive Bayes Algorithm", World Applied Sciences Journal, ISSN 1818-4952, 35 (1), pp:54-59, 2017. (SCOPUS Indexed Journal-Annexure II – Updated Sep 2016)

[13]. Nimala.K, Drmagesh. S, Thamizharasan. R," Performance Analysis Of Machine Learning Classifiers For Sentiment Analysis On Social Media Datasets", International Journal Of Pure And Applied Mathematics, Volume 115 No. 6 (2017) 597-603