Volume 5, Issue 8, August 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Performance Analysis of Supervised Learning Methodologies for Sentiment Analysis of Tweets

Gagandeep Singh, Kamaljeet Kaur Mangat
Department of Computer Science,
Punjabi University Regional Centre for Information Technology and Management
Punjab, India

Abstract— Every day, millions of people are using websites like Facebook, Twitter, Orkut, Google+ and many more to share their view in form of status updates, tweets, blogs etc. This material of social networking sites is used by companies to check the reviews of their product, which further help them for its betterment. For these online tweets, a new research area has been formulated that has been attracting the scholars to pursue their research i.e. Sentiment analysis. Sentiment analysis has grown to be one of the most active research areas in natural language processing. In this attitude, the objective of this research is to know about the performance of 4 supervised classifying algorithms for sentiment analysis of these tweets. A sentiment analysis framework was designed and implemented for this. Sanders twitter dataset has been used to implement this research. This dataset was refined using regular expressions. A Porter stemming and lemmatization algorithm has been used to refine the dataset for various anomalies. The dataset was divided in 60:40 ratio, in which 60% of dataset was used for training and 40% dataset was used for testing purposes. The results were analysed on parameters like - percentage of precision, percentage of recall, overall accuracy and kappa statistics.

Keywords— Sentiment Analysis, Tweets, Probability.

I. INTRODUCTION

Sentiment is said to be an organized system of emotional temperament cantered about the idea of some object. “Sentiment Analysis” is a rigorous computational review of person social interaction in the form of positive, negative or neutral sentiments.

Twitter is used by the user to convey their opinions and sentiments about some topic or object. The sentence that users post on the Twitter is known as “Tweets”. The tweets are generally written in 140 characters. More than 2 million people use the Twitter to convey their messages over the internet. This tends to the collection of large amount of data with some sematic meaning hidden in these tweets. Over the past few years this data has attracted many researchers to carry out their research in area of opining mining, review mining and sentiment analysis. This research is carried out to showcase the better supervised learning methodologies out of Naïve Bayes, Maximum Entropy, State Vector Machines and k-Nearest Neighbour for these tweets over the internet.

The outline of the paper is divided in parts. Section 2 describes the design of the sentiment analysis framework. Section 3 gives the outline of implementation of designed system. Section 4 gives the outline of performance metrics and carried results and paper ends with the conclusion in section 5.

II. DESIGN

The design of sentiment analysis framework is divided at various levels. The design of the framework is discussed in the figure 1. These modules are as follows:

1. Selection of tweets dataset.
2. Preprocessing of tweets
3. Reducing of Inflectional forms.
4. Training of refined tweets dataset using supervised learning classifiers
5. Testing of implemented sentiment analysis framework.

A. Selection of Tweets Dataset

This research utilized the tweets datasets available in public forums for training and testing. Two different datasets have been utilized for training and testing purpose. Sanders dataset provided by Niek Sanders on [1] is used for the training purposes. It consists of 5513 hand-classified tweets. Each entry in the dataset contains: Tweet id, Tweet text, Tweet creation date, Topic used for sentiment and Sentiment label: ‘positive’, ‘negative’, ‘neutral’ or ‘irrelevant’. It contains 573 positive tweets, 2503 negative tweets and 654 neutral tweets. The irrelevant tweets were removed from the dataset because of no usage in training.

B. Preprocessing of Tweets

The tweets in the dataset were refined for the anomalies like hash tags, @ tweeter names, webpage URL's, unwanted space and conversion of all the text in lowercase. To achieve this purpose the regular expressions (R.E.'s) were used. The dataset was refined using R.E.'s and saved in another dataset file name as Pre-Processed Tweets Dataset.

C. Reducing of Inflectional Forms

Inflection is the name for the extra letter or letters added to nouns, verbs and adjectives in their different grammatical forms. Nouns are inflected in the plural, verbs are inflected in the various tenses, and adjectives are inflected in the comparative/superlative [2]. These terms cause problems in the training of tweets. So, these inflectional forms in the Pre-Processed Tweets Dataset were removed using "Stemming" and "Lemmatization".

1. Stemming

Stemming is the procedure for decreasing inflected words to their original stem or base form. This stem is generally a written word form. The stem need not be identical to the morphological root of the word. It is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Stemming programs are commonly referred to as stemmers. The Porter Algorithm [3] has been used for the stemming purpose in our sentiment analysis system framework. The datasets are checked for the inflected words in the tweets and corrected in this part of design.

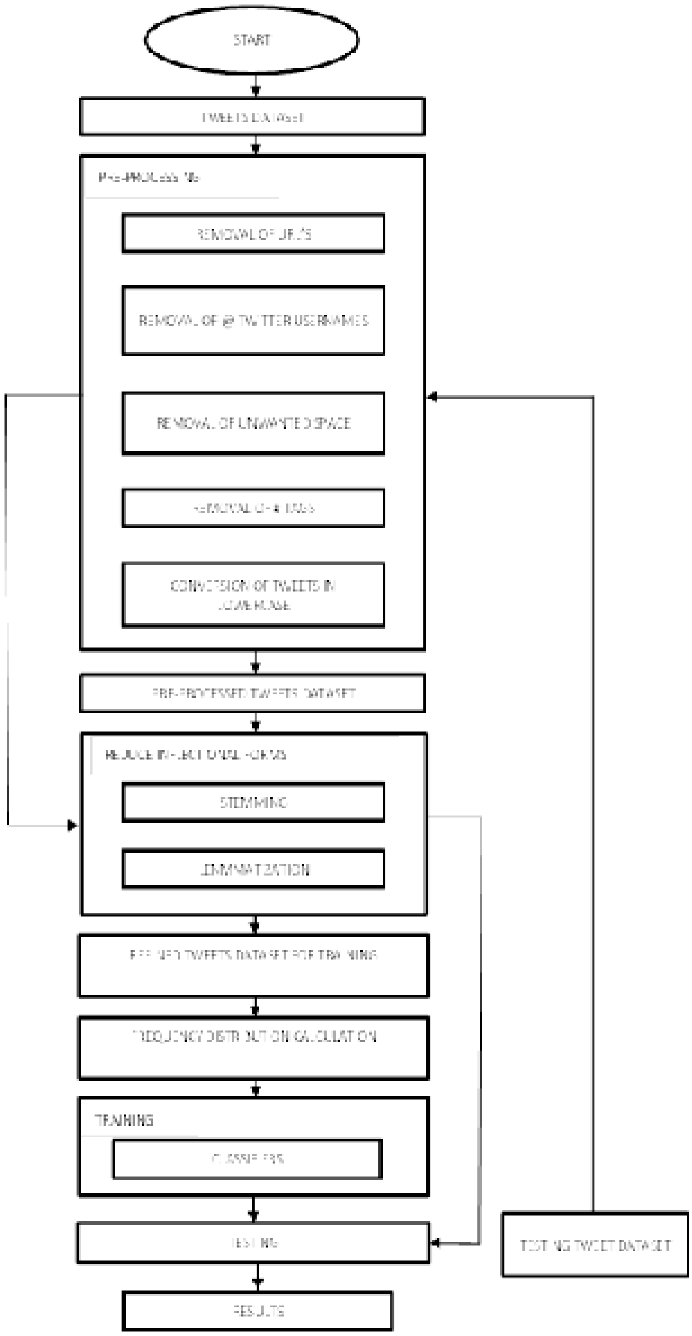

Fig 1: Design of sentiment analysis application framework

2. Lemmatization

In computational linguistics, lemmatization is an algorithmic process of determining the lemma for a given word. The process may involve complex tasks such as understanding context and determining the part of speech of a word in a sentence. Lemmatization is closely related to stemming. The difference is that a stemmer operates on a single word without knowledge of the context, and therefore cannot discriminate between words which have different meanings depending on part of speech. The tweets in the dataset are checked for the lemmas and lemmatization is performed on these words to make the dataset free from lemmas.

D. Training of Refined Tweets Dataset Using Various Algorithms.

Four classifiers Naïve Bayes, Maximum Entropy, Support Vector Machines and k-Nearest are used to train the network for classification of tweets in this sentiment analysis application. The descriptions of the classifiers used for supervised classification are explained below:

1. Naive Bayes Classifier

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features [4]. Given a class variable y and a dependent feature vector x through x_n, classification rule is defined as follows:

$$P(y|x_1, \ldots, x_n) \ \alpha \ P(y) \prod_{i=1}^{n} P(x_i|y) \qquad (1)$$

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i|y) \qquad (2)$$

2. Maximum Entropy Classifier

It is also known as logistic regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. The motive behind this classifier is that models are initialized with a uniform distribution and are updated as more information becomes available through training data. If certain event seems to be more likely occurred in the training data, then that event is weighted and the remaining probability mass is equally distributed across the rest of the distribution.

For example, suppose a distribution with four possible outcomes. If nothing is known about the events then this classifier would assign a probability of ¼ to each event.

If one of the events occurs ½ of the time but know nothing about the other three, it would assign a probability of ½ to the first event and 1/6 to each of the other three.

MaxEnt classifier belong to a family of classifiers known as log-linear classifier, which means they extract a set of features from the input and combine them linearly. Maximum Entropy classifier can be described in general using Equation 4.

$$p(c|d) = \frac{1}{z} exp\left(\sum_{i} w_i f_i\,(c,d)\right) \qquad (3)$$

Where c is a class and d is a given document. Z is a normalizing factor of the form

$$z = \sum_{c} p(c|d) = \sum_{c'=C} exp\left(\sum_{i=0}^{n} w_{c'i} f_i\right) \qquad (4)$$

which forces the weights to sum to 1, $f_i(c|d)$ is an indicator function learned from the training data and w_i is a weighting.

3. SVM Classifier

The fundamental concepts of support Vector machines were developed by Vapnik(1995) [5]. The SVM's notion is based on the idea of structural risk minimisation (SRM). SRM attempts to minimise the generalisation error i.e. the true error or unseen examples, which is bounded by sum of the training set error. Moreover, SVM do not suffer from over fitting as only the training data vectors that are needed to maximise the separation of the classes are used to define the decision boundary. These vectors are termed as support vectors. While there are many possible hyperplanes that could separate the vectors in a separable problem, the optimal hyper plane is one that separates the vector with maximum margin. If certain data vectors are added or removed from the training data, only those that are support vectors affect the decision boundary of the separating hyper plane. The model defined by the maximum margin hyper plane is based on the kernel function. Kernel function can be polynomial kernel function or other non-linear functions such as Radial Basis Function (RBF) kernel or a sigmoid function.

4. k-Nearest Neighbour Classifier

kNN is one of the simplest of classification algorithms available for supervised learning. The idea is to search for closest match of the test data in feature space. The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these. The number of samples can be a user-defined constant (k-nearest neighbor learning. The vectorized operations are used for computing distance. The common choice for similarity measures is Euclidean distance using equation (5).

$$distance = \sum_{i=1}^{N} (a_i - b_i)^2 = (a-b).(a-b \quad (5)$$

III. IMPLEMENTATION

The application was implemented on PC (Pentium(R) Dual-Core CPU T440 @ 2.20 GHz, 4GB RAM, Windows 8 and Ubuntu Platform, Python, Nltk Toolkit, Orange). The steps like preprocessing, stemming, lemmatization and frequency distribution calculation with classifying tasks performed are described in this section with its implementation.

A. Preprocessing Phase

Regular expressions and various NLTK inbuilt functions were used to achieve the preprocessing phase. In this phase various preprocessing steps like @ nametags, # tags, uneven distribution of white spaces and URL's removal mechanisms are discussed in this phase of sentiment analysis system. The steps used in this phase are discussed as follows:

1. Removal of URL's

The Twitter Users usually share the hyperlinks with prefix as https, ftp's and www's. The example of these tweets is discussed in figure 2**.** If we see from the point of text classification then these URL's is not that much important for training purposes. Regular expressions are used to detect and substitute these URL's with some text such as URL. Three working groups are made to achieve this task which is demonstrate in figure 3 and the same finite automata representation is represented in the figure 4.

Fig 2: Tweet example

1st Alternative: www\.[^\s]+

- www matches the characters www literally (case sensitive)
- \. matches the character . literally
- [^\s]+ match a single character not present in the list below
  Quantifier: + Between one and unlimited times, as many times as possible, giving back as needed.
- \s match any white space character [\r\n\t\f]

2nd Alternative: https?:[^\s]+

- 1st Capturing group (https?:[^\s]+)
- https matches the characters http literally (case sensitive)
- s matches the character s literally (case sensitive)
- [^\s]+ match a single character not present in the list below
  Quantifier: + Between one and unlimited times, as many times as possible, giving back as needed.
- \s match any white space character [\r\n\t\f]

3rd Alternative: ftp:[^\s]+

- ftp: matches the characters ftp: literally (case sensitive)
- [^\s]+ match a single character not present in the list below
  Quantifier: + Between one and unlimited times, as many times as possible, giving back as needed.
- \s match any white space character [\r\n\t\f]

Fig 3: Regular Expression capturing Groups representation for URL's Removal

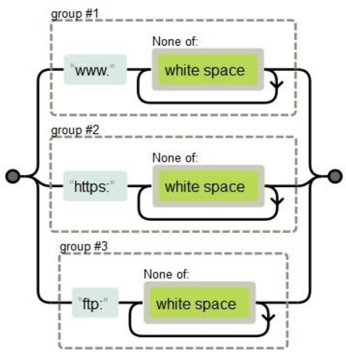

Fig 4: Finite automata representation of Regular Expression for URL's removal

2. Removal of @ user tags

When the twitter account is created, the twitter creates a username started with "@" symbol. In the preprocessing phase these username are removed with the help of regular expression. The capturing group and finite representation is shown in figure 5 and figure 6.

@[^\s]+

- @ matches the character @ literally
- [^\s]+ match a single character not present in the list below
  Quantifier: + Between one and unlimited times, as many times as possible, giving back as needed.
- \s match any white space character [\r\n\t\f]

Fig 5: Regular Expression Capturing Group representation for @ Removal

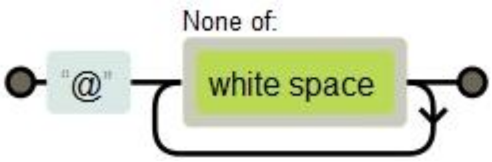

Fig 6: Finite automata representation of Regular Expression for @ Removal

3. Removal of Unwanted Space

Usually the user while using twitter handles type the message with lot of whitespaces. This whitespace character doesn't play any significant role in the twitter message and also dirty the dataset which was to be used to training purposes. In this purpose also the regular expression (RE) plays a significant role. The capturing group for RE which is used for the removal of unwanted space is shown in figure 7 and automata representation is also given in figure 8.

[\s]+

- [\s]+ match a single character present in the list below
  Quantifier: + Between one and unlimited times, as many times as possible, giving back as needed.

Fig 7: Regular Expression Capturing Group representation for Unwanted Space Removal

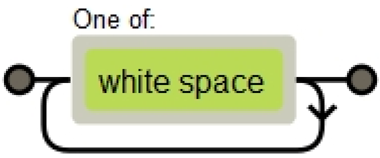

Fig 8: Finite Automata Representation of Regular Expression for Unwanted Space Removal

4. Removal of # Tags

A hashtag is a word or an un-spaced phrase prefixed with the hash symbol (#). These are used to both naming subjects and phrases that are currently in trending topics. For example, #iPad, #news. These hashtags also were removed with the help of RE. The capturing group and the automation representation are shown in figure 9 and respectively figure 10.

#([^\s]+)

- # matches the character # literally
- [^\s]+ match a single character not present in the list below
  Quantifier: + Between one and unlimited times, as many times as possible, giving back as needed.
- \s match any white space character [\r\n\t\f]

Fig 9: Regular Expression Capturing Group representation for removal of # Tags

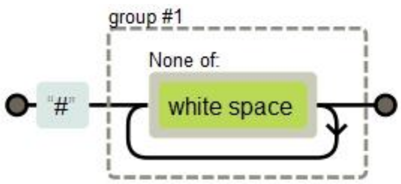

Fig 10: Finite automata representation of Regular Expression for Removal of # Tags

5. Conversion of Tweets in Lower Case

To make the training error free, the tweets were converted in the lower case characters so that for the training purpose the all the characters are equally treated by the training algorithms.

B. Implementation to Reduce the Inflectional forms

The Inflectional forms are removed with the help of porter and lemmatization algorithms which are discussed in this section.

1. Implementation of Porter Stemmer Algorithm

Martin Porter wrote a stemmer that was published in July 1980. This stemmer was very widely used and became and remains the de facto standard algorithm used for English stemming. It offers excellent trade-off between speed, readability, and accuracy. It uses a set of around 60 rules applied in 6 successive steps [3].

2. Implementation of Lemmatization

Lemmatization is the process of normalizing a word rather than just finding its stem. In the process, a suffix may not only be removed, but may also be substituted with a different one. It may also involve first determining the part-of-speech for a word and then applying normalization rules. It might also involve dictionary look-up. For example, verb 'saw' would be lemmatized to 'see' and the noun 'saw' will remain 'saw'. For our purpose of classifying text, stemming should suffice.

C. Frequency Distribution Calculation

Cumulative frequency is usually used to conclude the amount of comments that lie above or below to know the specific usefulness of comments or terms in a data set [6]. We have used the Unigram approach to for sending the dataset to the classifiers for training purposes. To know about the frequency of word that keeps repeating in the dataset. This cumulative frequency also helps us dodge having to balance the data, which can significantly cut the training time for refined tweets dataset. The cumulative distribution of top 50 words in our dataset is described using figure 11..

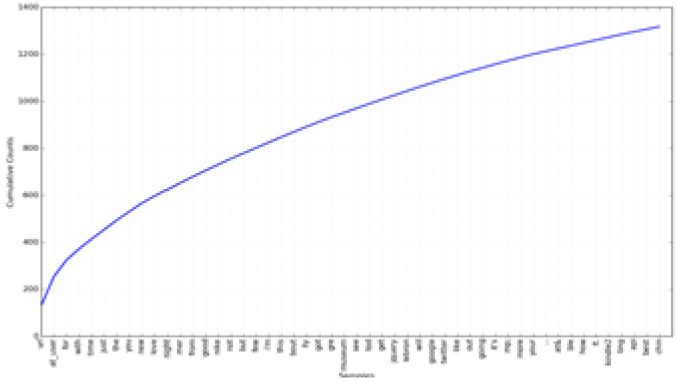

Fig 11: Cumulative Frequency Plot for 50 Most Frequent Unigrams in Refined Twitter Dataset

D. Training and Testing using Supervised Learning Methodologies

The refined twitter dataset is forwarded to the supervised mythologies in one by one manner for training purposes. In the same way, the testing dataset also goes from the same set of procedures applied to training dataset. The testing dataset is forwarded to the learned classifiers. The performance of the classifier is tested on various set of procedures and results are noted down at the end of system performance calculation.

IV. RESULTS

The performance of system is evaluated on the basis of various metrics and results are discussed. Section A describes various metrics used for performance evaluation. Section B discusses the results achieved for sentiment analysis of the refined tweets dataset.

A. Performance Metrics

The parameters selected for the evaluation of sentiment analysis of the refined tweets dataset are selected in such a way that effectiveness of the processes involved can be measured.

1. Percentage of Precision (P)

Precision measures the ratio of relevant output instances to the total instances obtained from output [7] . It is given as:-

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

Where tp is the number of true positives and fp is the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is neutral. In the proposed methodology precision can be calculated using equation 7.

$$Percentage\ of\ Precision = \frac{Correctly\ identified\ tweets\ in\ dataset}{All\ tweets\ retrieved\ as\ belonging\ to\ a\ seneiment\ class} \text{ X } 100 \quad (7)$$

2. Percentage of Recall (R)

Recall is ratio of relevant output instances to the total instances [7] moreover; it can also be represented as in equation 8.

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

Where *TP* is the number of true positives and *FN* the number of false neutrals. The recall is intuitively the ability of the classifier to find all the positive samples. This parameter is included to observe nature of output. In the proposed system Recall can be defined as in equation 9.

$$Percentage\ of\ Recall = \frac{Number\ of\ correctly\ identified\ tweets}{total\ number\ of\ tweets\ used\ for\ testing} \text{ X } 100 \qquad (9)$$

3. Overall Accuracy

The confusion matrix has been analyzed on the basis of overall accuracy. This assessment is used to find the total accuracy of sentiment analysis system. Overall accuracy sums up the total correctness of sentiment analysis mechanism of all the classifiers for which equation 10 and 11 has been used [8].

$$\omega = \sum_{i=1}^{nc} e_{ii}/_{NT} \qquad (10)$$

$$\text{NT} = \sum_{i=1}^{nc}\sum_{j=1}^{nc} e_{ij} \qquad (11)$$

Where, ω*:* denotes the overall accuracy to be calculated, *NT:* is the sum of all the off diagonal elements in the confusion matrix, e_{ii} is the sum of total numbers of correct cells along the major diagonal of confusion matrix, and nc: is total number of columns.

4. Kappa Statistics

Kappa coefficient is generally used as the measure of inter-rater reliability as overall accuracy is usually calculated along the major diagonal in confusion matrix but kappa takes all columns and rows in account while calculating [9]. Kappa ($\hat{k}$) has been used to find the overall reliability of sentiment analysis mechanism and the results were analysed on the standard of agreement for the kappa purposed by Landis and Koch (1977) (≤ 0 = poor, .01-.20 = slight, .21 - .40 = fair, .41 - .60 = moderate, .61 - .80 = substantial, .81 – 1 = almost perfect) [10]. The equation used to estimate the weighted kappa is:

$$\hat{k} = N\sum_{i=1}^{nc} X_{ii} - \sum_{i=1}^{nc}(X_{i+} \times X_{+i})/N^2 \; - \sum_{i=1}^{nc}(X_{i+} \times X_{+i}) \qquad (12)$$

Where, N: is total number of cell in matrix, nc: is total number of columns in confusion matrix, i+: is sum of column i, +i: is sum of row i, and X_{ii}: is total number of correct cells in matrix.

B. Experiment Results of Sentiment Analysis Framework

Performance analysis is discussed in this section for the implemented sentiment analysis framework. In our mention dataset total 3730 tweets. This dataset is divided in 60:40 ratio, in which the 60% of tweets were selected for training and 40% were selected for testing purpose. The selection of tweets is depicted in table 2.

Table 2: Tweets Dataset Division in 60:40 Ratios

	Total	**Training**	**Testing**
Positive	573	344	229
Neutral	2503	1502	1001
Negative	654	393	261

This table was analyzed using various classifiers implemented and is depicted using confusion matrix which is the also known as contingency matrix. The results were calculated on the basis of selected measures such as – percentage of precision, percentage of recall, overall accuracy and kappa statistics.

In table 3, the results of Maximum Entropy Classifier are displayed. The proposed classifier can predict 174 correct positive tweets out of 229 and 742 neutral tweets out of 1001 tweets. Also, the classifier is able to determine the total 203 negative tweets out of 261.

Table 3: Predicted Results for Maximum Entropy Classifier

		Predicted		
		Positive	**Neutral**	**Negative**
Observed	**Positive**	174	33	22
	Neutral	150	742	109
	Negative	18	40	203

In table 4, the results for Naïve Bayes classifier are discussed. The proposed approach could correctly identify 153 positive tweets 229, 720 neutral tweets out of 1001 tweets and total 182 negative tweets out of 261.

Table 4: Predicted Results for Naïve Bayes Classifier

		Predicted		
		Positive	**Neutral**	**Negative**

Observed	**Positive**	153	42	34
	Neutral	86	720	195
	Negative	3	76	182

In table 5, the proposed State Vector Machine is able to classify 187 positive tweets out of 229, 811 neutral tweets out of 1001 tweets and 205 negative tweets out of 261.

Table 5: Predicted Results for State Vector Machine

		Predicted		
		Positive	**Neutral**	**Negative**
Observed	**Positive**	187	39	3
	Neutral	14	811	176
	Negative	10	46	205

Table 6 states the results taken from the K-Nearest Neighbour which is able to classify 193 positive tweets out of 229, 851 neutral tweets out of 1001 tweets and 216 negative tweets out of 261.

Table 6: Predicted Results for K-Nearest Neighbour

		Predicted		
		Positive	**Neutral**	**Negative**
Observed	**Positive**	193	30	6
	Neutral	8	851	142
	Negative	7	38	216

V. DISCUSSION

The objective of the study is to find the finest applicable supervised classifying methodology out of our proposed classifiers - Naïve Bayes, Maximum Entropy, State Vector Machine and K-Nearest Neighbour Classifiers for sentiment analysis of tweets. The implementation of the Sentiment Analysis Framework was analysed using 4 validation techniques. Using these 4 performance metrics - Precision, Recall, Overall Accuracy and Weighted Kappa Statistics, the results carries out are analysed in this section.

Table 7 depicts the predicted precision for the 4 classifiers. The percentage of precision was analysed on difference levels of sentiments i.e. positive, neutral and negative sentiment. This clearly depicts that K-Nearest Neighbour perform best out of all the classifiers. But in if we tend to compare the prediction level of State Vector Machine and K-Nearest Neighbour at all 3 levels of sentiment analysis, the result are not that much fascinating as there is only less margin in the percentage of precision which usually don't make any great impact in this area of performance metric and perform same. This analysis is depicted in the form of graphical bar diagram in figure 12 respectively. The figure shows the percentage of precision for various implemented classifiers.

Table 7: Predicted Precision for Implemented Classifiers

Classifiers		**Naïve Bayes**	**Maximum Entropy**	**State Vector Machine**	**K-Nearest Neighbour**
Precision (%)	**Positive**	66.81	75.98	81.65	84.27
	Neutral	71.92	74.12	81.01	85.01
	Negative	69.73	77.77	78.54	82.75

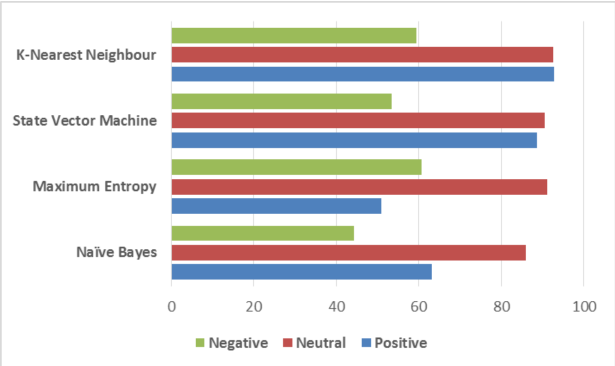

Fig 12: Depiction of Percentage of Precision in Implemented Classifiers

Table 8 depicts the predicted percentage of recall for implemented classifier for sentiment analysis. In this analysis also the K-Nearest perform best in class of the given classifiers in positive and neutral sentiment class. At neutral class of sentiment the maximum entropy, state vector machine and K-nearest are all most same as there is only 1% of difference in percentage of recall which usually doesn't matter. Figure 13 showcases the percentage of recall for various implemented classifiers.

Table 8: Predicted Recall for Implemented Classifiers

Classifiers		Naïve Bayes	Maximum Entropy	State Vector Machine	K-Nearest Neighbour
Recall (%)	**Positive**	63.22	50.87	88.62	92.78
	Neutral	85.91	91.04	90.51	92.6
	Negative	44.28	60.77	53.38	59.34

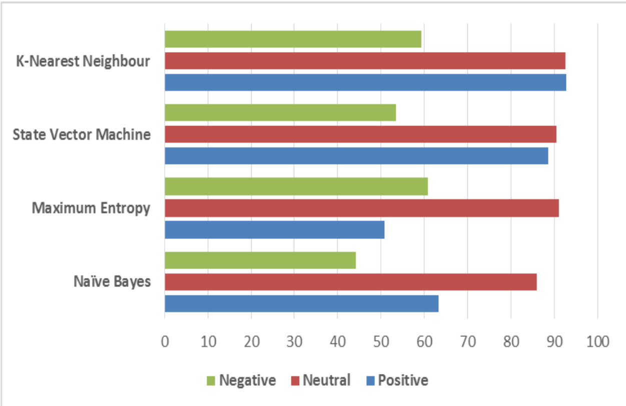

Figure 13: Depiction of Percentage of Recall in Implemented Classifiers

Overall accuracy in table 9 that was got after examination of sentiment for given classifiers, the K-Nearest Neighbour performs best in class with 84.5% overall accuracy. Figure 14 demonstrates the Overall Accuracy carried out in implemented Classifiers.

Table 9: Predicted Overall Accuracy for Implemented Classifiers

Classifiers	Naïve Bayes	Maximum Entropy	State Vector Machine	K-Nearest Neighbour
Overall Accuracy (ω)	70.75	75.05	80.68	84.5

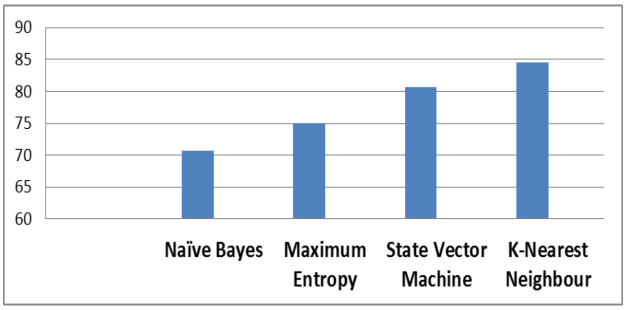

Figure 14: Depiction of Overall Accuracy in Implemented Classifiers

The kappa statistics ($\hat{k}$) in table 10 come out 0.46 in Naïve Bayes, 0.55 in Maximum Entropy, 0.63 in State Vector Machines and 0.7 in K-Nearest Neighbour Classifiers, which according to the Landis and Koch (1977) standard of agreement is fair for Naïve Bayes, moderate for Maximum Entropy, substantial for State Vector Machine and K-Nearest Neighbour Classifiers respectively. This clearly states that State Vector Machine and K-Nearest Neighbour Classifier perform equally well in this Sentiment classification task. The weighted Kappa Statistics is depicted using the line graph chart using figure 15.

Table 5.9: Predicted Weighted Kappa for Implemented Classifiers

Classifiers	Naïve Bayes	Maximum Entropy	State Vector Machine	K-Nearest Neighbour
Kappa ($\hat{k}$)	0.46	0.55	0.63	0.7

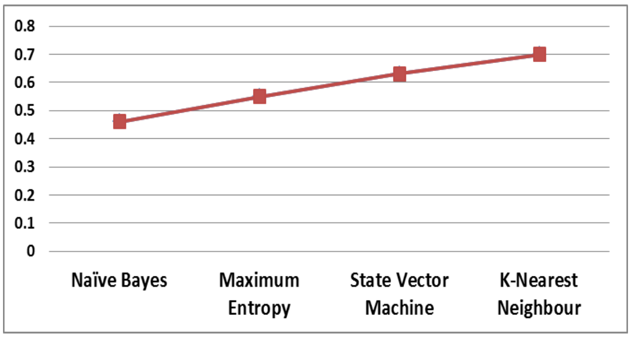

Figure 15: Depiction of Weighted Kappa Statistics in Implemented Classifiers

VI. CONCLUSION

This research emphasis on the sentiment analysis of tweets using supervised learning methodologies. This analysis of tweets is not only useful to e–commerce websites to know about the reviews about various products, but also product manufacturers. This research provides a comprehensive interpretation of the Sentiment Analysis task with great detail. As due to high importance of sentiment analysis in real world, the research in this field has been very effective in past time.

Experimental results clearly indicate that the suggested techniques are capable in executing their learning tasks. The experiment was performed using classifiers - Naïve Bayes, Maximum Entropy, State Vector Machines and k-Nearest Neighbour. Out of these classifiers State Vector Machines and k-Nearest Neighbour performed equally well with an accuracy of 80.68% and 84.5%.

REFERENCES

[1] N. Sanders, "Sanders Dataset." [Online]. Available: www.sananalytics.com/lab/twitter-sentiment/
[2] "Inflectional Term." [Online]. Available: http://esl.fis.edu/grammar/rules/inflections.htm
[3] M. F. Porter, "An algorithm for suffix stripping," *Progr. Electron. Libr. Inf. Syst.*, vol. 14, pp. 130–137, 1980.
[4] "Sklearn-NaiveBayes." .
[5] M. W. Corney, "Analysing E-mail Text Authorship for Forensic Purposes," Queensland University of Technology, 2003.
[6] "Cumulative Frequency." [Online]. Available: http://www.statcan.gc.ca/edu/power-pouvoir/ch10/5214862-eng.htm
[7] R. Jizba, "Measuring Search Effectiveness," 2008. .
[8] R. G. Congalton, "A review of assessing the accuracy of classification of remotely sensed data," *Remote Sens. Environ.*, vol. 37, pp. 35–46, 1991.
[9] J. Cohen, "A coefficient of agreement of nominal scales," *Educ. Psychol. Meas.*, vol. 20, pp. 37–46, 1960 [Online]. Available: http://epm.sagepub.com/cgi/doi/10.1177/001316446002000104
[10] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data.," *Biometrics*, vol. 33, pp. 159–174, 1977.

