

# Opinion Mining: An Approach to Feature Engineering

Shafaq Siddiqui<sup>1</sup>, M. Abdul Rehman<sup>2</sup>, Sher M. Daudpota<sup>3</sup>, Ahmad Waqas<sup>4</sup>

Department of Computer Science  
Sukkur IBA University  
Sukkur, Sindh, Pakistan

**Abstract**—Sentiment Analysis or opinion mining refers to a process of identifying and categorizing the subjective information in source materials using natural language processing (NLP), text analytics and statistical linguistics. The main purpose of opinion mining is to determine the writer’s attitude towards a particular topic under discussion. This is done by identifying a polarity of a particular text paragraph using different feature sets. Feature engineering in pre-processing phase plays a vital role in improving the performance of a classifier. In this paper we empirically evaluated various features weighting mechanisms against the well-established classification techniques for opinion mining, i.e. Naive Bayes-Multinomial for binary polarity cases and SVM-LIN for multiclass cases. In order to evaluate these classification techniques we use Rotten Tomatoes publicly available movie reviews dataset for training the classifiers as this is widely used dataset by research community for the same purpose. The empirical experiment concludes that the feature set containing noun, verb, adverb and adjective lemmas with feature-frequency (FF) function perform better among all other feature settings with 84% and 85% correctly classified test instances for Naive Bayes and SVM, respectively.

**Keywords**—Opinion mining; feature engineering; machine learning; classification; natural language processing

## I. INTRODUCTION

Sentiment analysis or opinion mining is a process of recognizing and categorizing people’s sentiments, opinions, attitudes, and emotions from the text written in natural language. Because of the proliferation of text data on web and social media, opinion mining has gained a lot of attention and has in this way turned into an active research area in natural language processing NLP, which exploits systems and techniques from data mining.

Every hour, millions of messages are posted on social media like twitter, rotten tomatoes and Facebook. These messages cover numerous topics including public opinion about various topics such as products, current affairs, politics, and movies and so on. The public opinions or sentiments against a product in fact impact the market trends. For instance, [1] found a strong correlation between positive sentiments on twitter and the box-office collection of the movies. The polarity ration regarding a movie has an influence on movie revenue. For example, in the first week of release of “*New Moon*” movie polarity ratio was 6.29 and box-office collection was 142M. In the second week the ratio dropped to 5 and the box-office collection also dropped to 34M. The polarity ration of tweets can be measured as under:

$$PNratio = \frac{|Tweets\_with\_positive\_sentiments|}{|Tweets\_with\_negative\_sentiments|}$$

In the opening week of “*The Blind Side*” movie, the box-office collection as 34M with the polarity ratio of 5.1 and then in the second week polarity ratio increased to 9.61 and the collection also increased to 41M.

Recently, the increased demand of employing opinion mining for decision making in various application domains has drawn a considerable attention of research community from computer science towards the development of practical solutions. For instance, in [2] authors reported a strong correlation between public mood on social media and political as well as cultural events like the Presidential Election and Thanksgiving Day. There are a lot of example of real world use cases where sentiment analysis has been exploited for decision making [2, 3,4]. Thus, many algorithms and methods for sentiment analysis evolved in recent years [5]. In order to get deeper understanding into strengths and shortcomings of these methods, it is important to evaluate them in different settings with a variety of preprocessing choices.

In this paper, we empirically performed the evaluations of two well established classification methods, i.e. Naive Bayes-Multinomial for binary polarity cases and SVM-LIN for multiclass cases. These methods are evaluated by means of three different feature settings in preprocessing phase. The feature settings include Feature Presence (FP), which represents binary values, Feature Frequency (FF) and Term Frequency–Inverse Document Frequency TF-IDF both represent real values. We have validated the evaluation using publicly available dataset of movie reviews taken from Rotten Tomatoes. The empirical evaluation revealed an interested conclusion that the feature-frequency (FF) setting performed better among all other feature settings with 84% and 85% accuracy for Naive Bayes and SVM, respectively.

The rest of the paper is organized as follows. The background section contains some related and technical knowledge about the domain. The Literature Review section contains previous research which relates to the experiment. The Methodology section introduces the specifics of the experiment. The Results section presents the results of the work. Lastly, the Conclusion section presents the conclusion.

## II. BACKGROUND

### A. Natural Language Processing

Natural Language Processing (NLP) is a framework to support an interaction between computers and human (natural) languages by providing processing capability of a text written in natural language using the methods and techniques stemming from various fields like computer science, computational linguistics and artificial intelligence. Natural language processing provides us various algorithms for understanding and recognizing the patterns of human language especially statistical algorithms, which are based on machine learning. The machine-learning algorithms learn different rules through the analysis of large corpora (A hand-annotated documents with their respective polarity values to be learned by algorithm) of typical real-world. These algorithms take a large set of features generated from corpora as input. Research in natural language processing is now focused towards soft and probabilistic predictions based on assigning weight to all features. Such models have an advantage of expressing the relative certainty of many different possible answers rather than only one, thus they provide more reliable and accurate results when such kind of a model is included as a component of larger systems some of these algorithms are Naive Bayes, Maximum Entropy Measure and SVM [6].

### B. Finding Appropriate Features

Sentiment Analysis is a task of performing text classification. In supervised classification different machine learning algorithms can be used to classify the text. In supervised learning the major focuses are features selection and choosing appropriate classification algorithm. In Natural language processing the terms features and token are used interchangeably. Finding best features are very important when text mining is performed using machine learning algorithms. Features which tend to be consistent in text of a certain class are generalized as a good indicator of that class [6, 8]. For example the word bad may be a good indicator to identify a text as negative. However, many features such as unigrams, bigrams, trigrams, POS tagged unigrams, dependency trees and several other have been used in sentiment analysis [1, 2, 5]. The purpose of finding these features is to find good indicators to generalize for text classification.

Some of the several feature types are discussed below:

- N-grams

We use n-grams to capture the dependencies between all words which appear in a sentence structure sequentially; n-grams combination does not preserve the words' syntactical or semantic relations. An n-gram is a probabilistic language model which predicts next word conditioned on the occurrence of previous word. The probabilistic expression is  $P(x_i|x_{i-(n-1)} \dots, x_{i-1})$ .

N-grams can be of size 1 (unigrams) size 2 (bigrams), size 3 (trigrams) and so on.

Finding of Pang et al. (2002) research states that unigrams perform better than bigrams [10, 11].

- Parts of Speech Tagging

POS tagging has been used for a long time in text classification and Natural Language Processing (NLP). POS tagging differentiates syntactic meaning of words in a sentence by using some specific tags, such as tags for noun, pronoun, verb, adjective, adverb, conjunction and others.

In sentiment analysis POS tagged words are used as features for classification as the adjective can provide good clues about the polarity of the sentence. In 2011 Mejova et al. conducted a study in which they tested POS tagged features effectiveness in supervised learning separately and with combinations of POS tags. The combinations of adjectives, adverbs and nouns performed better than other combinations when treated as features and all individual POS tagged features were outperformed by adjective when used as features. In this study we have applied Apache OPENNLP Maxent POS tagger model. Some POS tags are defined in Table 1 below.

- Syntactic Dependency Tree Patterns

A syntax dependency tree is a syntax tree structure that captures the dependency between a word (root) and its dependents (Childs) it identifies useful semantic relationships. In syntax tree the relation among nodes are based on their grammatical dependency. Dependency parsing identifies parts of speech and syntactic relations and then determines the grammatical structure of sentence. Many researches have been conducted for determining the efficient and accurate parsing tree pattern for sentiment analysis.

Except that appropriate feature selection, assignment of numerical feature values to selected feature is also important. This value assigning method is called the weighting method and most widely used weighting methods are term frequency (TF) and presence.

### C. Feature Selection Methods

Feature selection methods are techniques to reduce the size of features space and to choose small set of features to capture relevant properties or classification of dataset.

An effective feature selection method can increase the efficiency of classifier.

TABLE I. POS TAGS

Tag	Parts of Speech
JJ	Adjective
NN	Noun, singular or mass
RB	Adverb
VB	Verb, base form
NNS	Noun, Plural

Many feature selection method has been proposed such as Information Gain (IG), Mutual Information (MI), X2 test (CHI), term strength (TS) and term presence.

Normally the size of feature vectors for a document and sentence is usually bif specially when unigram features are used and these large sized vectors can slow down the system performance one way to get rid of these large vector which contains less efficient features is to pre-process the data such as removal of stop word

#### D. Naive Bayes Classifier

Bayesian classifier, a statistical technique, predicts the probability of an event to belong to a particular class. The classifier is based on Bayes Theorem [7] which states that the probability of an event is based on the prior (conditional) knowledge of the event. It can mathematically be stated as under:

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)} \quad (1)$$

where

A and B are events and  $P(B) \neq 0$ .

$P(A)$  and  $P(B)$  are the probabilities of A and B without considering their conditional interaction.

$P(A | B)$  is a conditional probability of event A given a condition that the B is true.

$P(B | A)$  is a conditional probability of event B given a condition that the A is true.

given sample to belong to a particular class.

#### E. Support Vector Machine Classifier

Support Vector Machines are based on the idea of decision planes for defining decision boundaries. We used decision planes for separating the objects having different class memberships. A decision plane creates a boundary between them.

### III. LITERATURE REVIEW

Due to the rapid growth of text data on web and social media, opinion mining has gained a lot of attention and has thus become an active research area in natural language processing, which exploits techniques and methods from data mining. The sentiment Analysis is not only limited to computer science domain but it has also spread to management sciences and social sciences application domains due to its importance and applications to business areas and society. In fact, the importance of opinion mining is proportional to the growth of social media such as Facebook, social discussion forums, Twitter and product review websites. With the growth of social media we now have huge stores of raw datasets which can be analyzed to find different informative patterns [8, 9].

A simple use case for sentiment analysis is to discover what people feel about a particular topic and what their attitude about a particular topic is. For example:

Do people on a social chat group think that the recently released movie was a block buster or flop one?

The newly opened restaurant is serving best sea foods?

What is the public opinion for a particular election candidate?

By analyzing tweets for sentiments will answer these questions. Furthermore, we can also learn why people think that the movie was a hit or flop by extracting the exact words indicating why people did or didn't like the movie. For example, poor plot, and or bad casting.

This is the kind of insight one hopes to find when conducting market research. Now one can easily decide that in which particular direction he/she need to work more in next film either on plot or cast.

From many years research has been focusing on different levels of classification either document level classification, sentence level classification or phrase level classification using supervised or unsupervised learning. For supervised learning selection of features, feature weight assignments and features selection methods play an important role in classification performance.

Prior to the text polarity classification which identify that the document as positive or negative several research studies were conducted for subjectivity classification of document that whether the document or sentence is subjective or objective?

In 2002, Pang and lee conducted a sentiment analysis study using movie review data. In that document level supervised learning they classify the document using Naïve Bayesian, SVM and Maximum Entropy. They choose several token such as POS tags, adjectives and n-grams as features and they found that the machine learning methods outperformed the human classification (they asked their two students to classify the documents). They also found that SVM outperformed other machine learning algorithms and unigrams perform better than bigrams [10].

In 2004, Bo Pang and Lillian Lee again conduct a study in which they perform Sentiment Analysis Using Subjectivity Analysis Based on Minimum Cuts [11]. In this study they examined the relationship between subjectivity detection and polarity classification. Their findings show that text subjectivity detection can compress it into small and much shorter extracts but retaining polarity information at a comparable level to that of full review.

Their work identifies that subjectivity extracts classified by Naïve Bayes are more effective inputs. Their work also shows that utilizing context based information via minimum-cut framework can see statistically significant improvement in

polarity-classification accuracy. In [12], Eguchi et al. proposed methods in which he define some assumptions that sentiment expressions are related to their topics. As for example, a negative view for a politician may be indicated using reckless or negative review for a voting event may be indicated by Flaw. In their research Eguchi combined topic relevance models and sentiment relevance models. They create a training dataset by annotating S (sentiment) and T (topic) to sentiment. Then these S, T and polarities of sentiment together formed a triangular relationship and they obtained high performance.

In [13], Riya et al. introduce a hybrid approach on Sanders analytics dataset. The classification was a combination of both Knowledge base and machine learning approach in which each word was first classified using knowledge based approach with the help of SentiWordNet then the complete tweets was classified using different classifiers. The hybrid approach results in 100% accuracy, the Naïve Bayes classifier with a total accuracy of 75%. The paper concludes that in sentiment classification the machine learning techniques are easier than symbolic techniques.

The approach of Tirath, Sanjeev [14] was more focused on calculating some robust features. The features having information gain (IG) score greater than zero where considered for classification.

Asha et al. [15] used three different dataset to report the accuracy of Naïve Bayes and SVM classifiers. The features selection approach was based on the extraction of TF, TF-IDF, opinion oriented keywords using SentiWordNet. The features were then assigned a weight using GINI index. The results show that both classifiers performed better on large movie review data set SAR14.

Dhiraj Gurkhe's [5] dataset was an amalgamation of tweets, movie reviews dataset, hand classified tweets from Sanders, emoticons dataset and sentiment lexicons. Three different features vectors were used for classification having unigrams, bigrams and unigrams+bigrams. The classification results show that the Naïve Bayes performs better by using unigrams features.

#### IV. METHODOLOGY

In this study an experiment on Rotten Tomatoes movies reviews dataset was performed. This dataset contains 1500 positive reviews and 1500 negative reviews. The purpose behind choosing the movie reviews dataset is that they are more detailed and often considered as good material for subjectivity and polarity classification. Typical comments are usually very short such as tweets that are only one or two sentences long. These comments are narrowly focused on a single topic of interest expressed. Whereas movie reviews tend to be more detailed and focused on whole story, acting, actors and give an overall impression about the movie.

For performing machine learning, the focus is to find some relatively correct clues from the text which can lead to correct classification. These clues about the original data are called features and are stored as a feature vector,  $F = (f_1, f_2, \dots, f_n)$

in feature vectors each coordinate represent one clue say feature  $f_i$  of the original text.

Features selection strongly influences the classifier learning. In feature selection this study goal is to capture desired properties of text in some numerical form. The choice of features is based upon their relevancy with sentiment analysis task. The algorithms for extracting best feature sets does not exist, thus we only rely over research intuitions, expertise in field, domain knowledge and performing various experiments for choosing the best set of features [17, 18].

In this study we have focused on unigrams and we used Apache OPENNLP an open-source java library for extracting the relevant linguistic features from the corpora. OPENNLP is a set of classifiers which work on word level. As we are working on word level we removed all the punctuations and emotions from the text. Table 2 below summarizes the types of word level features we used for classification. We experiment with different combination of above mentioned features and along with these features we have also used three different weighting functions and then choose the feature set and weighting function which performed best for the Sentiment Analyzer.

##### A. Negation Handling

Negation handling is an important part of polarity analysis. Some of the sentences such as "it was not a good movie" has the opposite polarity from the sentence "It was a good movie". Word influenced by the negation especially adverbs and adjectives should be treated differently. One way to handle negation is to use a bigram dictionary including special feature word NOT for every adverb and adjective [16].

Another way could be to perform parsing of all sentences, but this approach is computationally expensive and may cause inaccuracies if the corpus is not completely tagged. Another approach is to construct training dataset having all possible negation sentences, but this requires time and efforts to construct an optimal dataset.

In this approach we have dealt with some simple cases of negation such as not, do not, doesn't. We have performed POS tagging and have defined some rules for checking the pre-determiner of adverbs and adjectives this approach has increase the performance a little better, it would possible to define more extensive rule for more better performance that would deal with noun and verbs instead.

TABLE II. FEATURES USED FOR CLASSIFICATION

Feature	Definition	Example
Word	All words in text	Come, world
Lemma	Word entry in dictionary	Watching = watch children = child, players = player
Parts of Speech (POS) tags	Noun, Verb, Adjective	The girl was very <b>beautiful</b> In this sentence the word <b>beautiful</b> is adjective

### B. Weighting Functions

The three weighting functions we used in this experiment are:

1) *Feature presence (FP)*: It represent a binary value, these binary values indicate the absence and presence of a feature in text (e.g. in the text–“good day”, only the features “good” and “day” are set to 1) and all the other remaining words in the vocabulary (set of words we see in corpus) are set to 0.

2) *Feature frequency (FF)*: It represents a real value, which indicates the occurrences (frequency) of a feature in a given example, the frequency value is normalized according to the size of the text (in words).

3) *TF-IDF*: It represents a real value, which indicates the occurrences (frequency) of the feature in a given text; this frequency value is then divided by the logarithm of the number of examples from the corpus containing this feature. This can be explained as for features  $f_i$ ,

$$TF - IDF(f_i) = \frac{FF_i}{\log(DF_i)} \quad (2)$$

Where  $FF_i$  denotes feature frequency of  $f_i$ , and " $DF_i$ " denotes the document frequency of  $f_i$  (number of documents containing  $f_i$ ). The purpose of using this weighting function was to give a larger weight to features that were seen less in the corpus than the common ones. In other words, we increase the impact of rare words over common words.

### C. Classification

After determining the best features we performed classification. In performing classification task each sentence is considered as conditionally independent. In this study we performed supervised classification. Classifier learning or training is done using cases from training set and later the quality of training is evaluated using the cases from the test set. The labels of interest are polarity labels “Positive” and “Negative” for Naïve Bayes and SVM classifier.

We have taken the following steps for performing classification:

1) *Data preprocessing*: Training dataset was preprocessed by applying POS tagging, location and subject tagging, removing stop words and punctuations.

2) *Feature extraction*: Suitable features were extracted for classification using different combinations of word level features and weighting functions discussed above.

3) *Model building*: Using features the classifiers were trained and a model was created.

4) *Model evaluation*: The classifiers were evaluated by mean of confusion matrix and ROC analysis.

5) *Classification*: The test cases were classified using the classification models. The test dataset for classifiers had 1500 positive and 1500 negative reviews. The algorithm 1 describe our approach to data cleaning and training dataset preparation before performing classification.

### D. Algorithm

Algorithm 1: Training dataset preparation

```
1. Start [input: raw dataset with labels, output: training dataset (features with labels)]
2. foreach review r in dataset do
3.   stemming(r)
4.   stopping(r)
5.   case_normalization(r)
6.   POS_tagging(r)
7.   tokenize(r)
8.   foreach token t in r do
9.     IF t is not subjective & t ∈ {verb, adverb, adjective and noun} then
10.      feature ← feature U t
11.    END IF
12.  END LOOP
13.  write(feature, label)
14. END LOOP
15. END
```

## V. RESULTS AND DISCUSSION

In this experimental setup we try to see how each classifier perform on the dataset when using different features settings. We have defined the five different feature settings for each of the weighting function. These settings are defined in Table 3.

We have compiled the results for all five different features setting with all three different weighting functions for both classifiers Naive Bayes and SVM.

Fig. 1, 2 and 3 shows the result (Percentage of correctly classified test instances) of running Naïve Bayes and SVM classifier on testing dataset with weighting functions FP, FF, TF-IDF, respectively.

TABLE III. FEATURE SETTINGS

Features Settings	Description
Unigram (U)	Using words without subject words
Lemma (L)	Using the lemmas from WordNet Subject words are not considered
Lemma + POS tags (L+P)	All lemma concatenated with their respective parts-of-speech tag. Such as (beautiful_adjective). Subject words are not considered
Selected Lemma (SL)	Only selected lemmas whose parts of speech are adjective and adverb because it has been observed that the subjective sentiments of an author are reflected mostly on adverbs and adjectives. Subject words are not considered
Selected Lemma 2 (SL2)	Only selected lemmas whose parts of speech are adjective, adverb, noun and verb so that the dependencies can be captured. Subject words are not considered

Fig. 1 depicts the performance graph of both SVM and Naïve Bayes algorithm using different features and feature presence weighting function. The graph show that when adjective, adverbs, nouns and verbs were extracted as features the SVM classifier outperformed the Naïve Bayes classifier with overall accuracy of 85%, when tested on test dataset while the Naïve Bayes classifier gave 84% accuracy.

Fig. 2 depicts the performance graph of both SVM and Naïve Bayes algorithm using different features and feature frequency weighting function. The graph show that when adjective, adverbs, nouns and verbs were extracted as features the SVM classifier again outperformed the Naïve Bayes classifier with overall accuracy of 85.5%, when tested on test dataset while the Naïve Bayes classifier gave 84.5% accuracy. The weighting function feature presence increased the performance of both classifiers.

Fig. 3 depicts the performance graph of both SVM and Naïve Bayes algorithm using different features and Term frequency-inverse document frequency weighting function. The graph show that the SVM gave an accuracy of 83% when adjective, adverbs, nouns and verbs were extracted as features but the performance of Naïve Bayes classifier depletes badly. Naïve Bayes classifier gave 83% accuracy when lemmas were extracted as features.

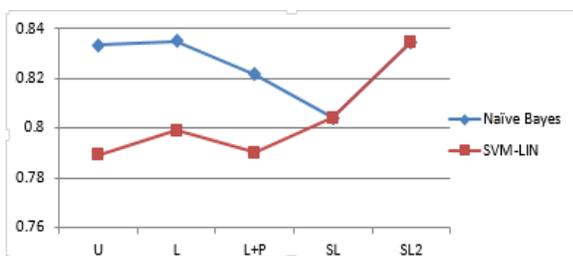


Fig. 1. Accuracy Results using Test Dataset with TF-IDF.

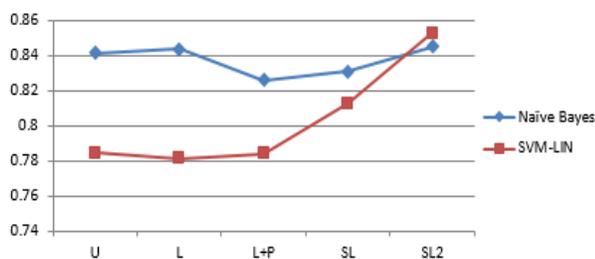


Fig. 2. Accuracy Results using Test Dataset with Feature-Frequency (FF).

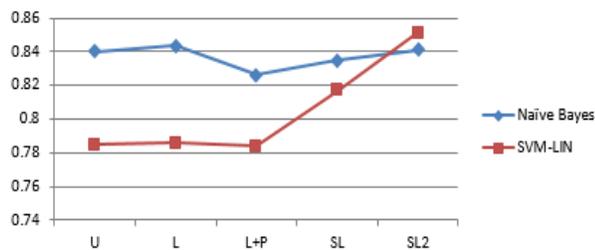


Fig. 3. Accuracy Results using Test Dataset with Feature-Frequency (FF).

TABLE IV. CONFUSION MATRIX

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	1279	221	Positive Predicted Values	0.852
	Negative	203	1297	Negative Predicted values	0.864
		Sensitivity	Specificity	Accuracy = 0.858	
		0.86	0.854		

After analyzing the results of the experiment we can conclude that the features setting SL2 which includes only adverb, adjective, verb and noun lemmas perform good using all three weighting functions when used by both classifiers.

As it is now clear that the models build using feature set of adverb, adjective, verb and noun lemmas performed best when used with feature-frequency weighting functions with accuracies 84% and 85% for Naïve Bayes and SVM-LIN respectively we evaluate the models using a test dataset of 1500 positive, 1500 negative movie reviews. The confusion matrix in Table 4 shows the positive predicted rate, negative prediction rate, sensitivity, specificity and accuracy of Naïve-Bayes classifiers. The F-score of test results was 0.85.

The true positive rate of the build was 86% while the true negative rate was 85%. The overall accuracy achieved was approximately 85%. The performance of both classifiers was also evaluated using ROC curve analysis. The ROC curve analysis is shown in Fig. 4 and 5. The ROC curve shows an excellent classification test accuracy for Naïve Bayes classification model with value 0.93 while the ROC curve accuracy for SVM classification model was 0.72.

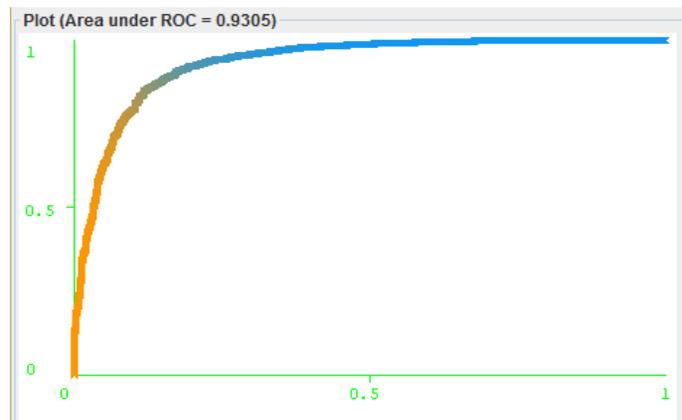


Fig. 4. ROC Curve of Naïve Bayes Classifier.

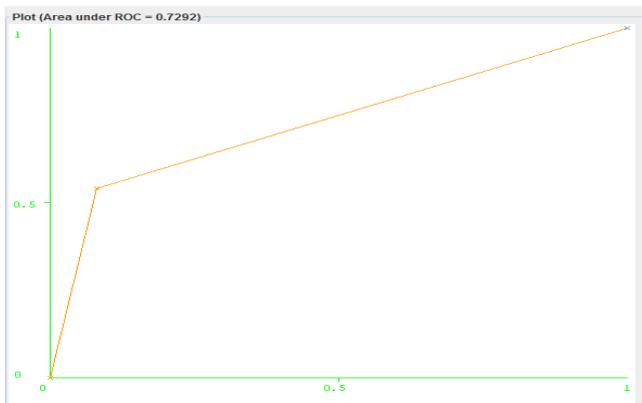


Fig. 5. ROC Curve of SVM Classifier.

## VI. CONCLUSION

We have performed classification on two different datasets and calculated simple word based features for classification. To find most appropriate feature set we defined five different feature settings and use three different weighting functions and then calculated the accuracies of features for each feature setting and identified that the feature set containing noun, verb, adverb and adjective lemmas with feature-frequency (FF) function perform better among all other feature settings with 84% and 85% correctly classified test instances for Naïve Bayes and SVM, respectively.

## REFERENCES

- [1] Bollen, Johan, Huina Mao, and Xiao-Jun Zeng. (2011) Twitter mood predicts the stock market, *Journal of Computational Science*
- [2] Asur, S., & Huberman, B. A. (2010, August). Predicting the future with social media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01* (pp. 492-499). IEEE Computer Society.
- [3] A. Mittal, A. Goel, (2011) Stock prediction using Twitter sentiment analysis, Stanford University, CS229
- [4] Diakopoulos, N. A., & Shamma, D. A. (2010). Characterizing debate performance via aggregated Twitter sentiment. In *Proceedings of the 28th international conference on Human factors in computing systems* (pp. 1195-1198). New York, NY: ACM Press.
- [5] Dhiraj Gurkhe, Niraj Pal, Rishit Bhatia. (2014). Effective Sentiment Analysis of Social Media Datasets. *International Journal of Computer Applications*, 4.
- [6] Julian Brooke, 2009, A Semantic Approach to Automated Text Sentiment Analysis. Springer-Verlag Berlin Heidelberg, 478-483.
- [7] Walaa Medhat, Ahmed Hassan, Hoda Korashy: Sentiment analysis algorithms and applications: A survey, *Ain Shams Engineering Journal*, Elsevier, p. 1093-1113, Volume 5, Issue 4, December 2014
- [8] J. Han, M. Kamber, and J. Pei. 2006 *Data mining: concepts and techniques*. Morgan kaufmann series,
- [9] Yessenov K, Misailovic S. (2009). Sentiment analysis of movie review comments, 1-17 <http://people.csail.mit.edu/kuat/courses/6.863/report.pdf>.
- [10] Pang, B. and Lee, L, 2004, A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts.
- [11] Pang, B., Lee, L., and Vaithyanathan, S, 2002, Thumbs up? Sentiment Classification Using Machine Learning Techniques.
- [12] Koji Eguchi, Victor Lavrenko. (2006). Sentiment retrieval using generative models, *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia
- [13] Riya Suchdev, Pallavi Kotkar, Rahul Ravindran, Sridhar Swamy. (2014). Twitter Sentiment Analysis using Machine Learning and Knowledge-based Approach. *International Journal of Computer Applications*, 36-40.
- [14] Tirath Prasad Sahu, Sanjeev Ahuja. (2016). Sentiment Analysis of Movie Reviews: A study on feature selection and classification algorithms. *IEEE*.
- [15] Asha S Manek, P Deepa Shenoy, M Chandra Mohan, Venugopal., (2015). Aspect term extraction for sentiment analysis in large. Springer Science.
- [16] Akshay Amolik, Niketan Jivane, Mahavir Bhandari, Dr.M.Venkatesan. (2015). Twitter Sentiment Analysis of Movie Reviews using Machine Learning Techniques.
- [17] M. K. Elhadad, K. Badran and G. I. Salama, "A novel approach for ontology-based dimensionality reduction for web text document classification," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pp. 373-378, Wuhan, China, 2017.
- [18] Mazandu, G. K., Chimusa, E. R., & Mulder, N. J. (2016). Gene ontology semantic similarity tools: survey on features and challenges for biological knowledge discovery. *Briefings in bioinformatics*, 18(5), 886-901.