

# Proposed Approach for Sarcasm Detection in Twitter

Shubhodip Saha, Jainath Yadav and Prabhat Ranjan\*

Department of Computer Science, Central University of South Bihar, Patna – 800014, Bihar, India;  
shubhodip@cusb.ac.in, jainath@cub.ac.in, prabhatranjan@cub.ac.in

## Abstract

**Background:** Sarcasm detection in twitter is a very important task as it had helped in the analysis of tweets. With the help of sarcasm detection, companies could analyze the feelings of user about their products. This is helpful for companies, as the companies could improve their quality of product. **Methods:** For preprocessing of data TextBlob is used. TextBlob is a package installed in Natural Language Toolkit. The preprocessing steps include tokenization, part of speech tagging and parsing. The stop words are removed using python programming. The stop words corpus which consist of 2400 stop words and which is distributed with NLTK have been used. RapidMiner is used for finding polarity and subjectivity of tweets. TextBlob is used for finding the polarity and subjectivity confidence. Weka is used for calculating the accuracy of tweets based on Naïve Bayes classifier and SVM classifiers. **Findings:** The paper provides the polarity of tweets which include whether the tweet is positive, negative or neutral. Polarity confidence and subjectivity confidence are also found. Accuracy of tweets are found using Naïve Bayes and SVM classifiers. **Applications:** Sarcasm Detection could be helpful in analyzing the exact opinion of the user about a certain thing.

**Keywords:** Polarity, Polarity Confidence, Sarcasm Detection, Subjectivity, Subjectivity Confidence

## 1. Introduction

Twitter have become the largest platform where people express their feelings, opinions, view and real time events such as live tweets etc. With respect to previous years, the data of twitter has increased much and thus forming big data. Twitter has 315 million monthly active users, eighty two percent of active users on mobile and millions of tweets are being circulated through twitter every day. Various organizations as well as companies are interested in twitter data for finding the views of various people towards their products or events. Twitter is also used to find out various views of people towards political events, movies etc.

But due to 140 characters per tweet limit as well as the non formal language such as slang, emoticons, hash tags etc. used in tweets, finding views of different people is very difficult. To detect sarcasm among those tweets is comparatively more difficult. Sarcasm refers to expressing negative feelings using positive words. Oxford dictionary

explains sarcasm as “the use of irony to convey contempt”. Macmillan dictionary explains sarcasm as “activity of saying or writing opposite of what the speaker wants to mean or the way of speaking intended to make someone feel stupid or angry”. Collins dictionary explains sarcasm as “mocking or ironic language to convey insult or anger”. Sarcasm and irony are considered two very close concepts. Sarcasm is also when people mean something else from what he speaks. Sarcasm is used not only to make jokes but also for criticizing other people, views, ideas etc. As reason of which sarcasm is very much used in twitter. For example-”I loved being ignored”. Here “love” expresses a positive sentiment in negative context. Therefore this tweet is referred as sarcastic. Thus analyzing sarcastic tweets is very difficult.

Three types of sarcasm have been shown by<sup>1</sup> which are following:

1. Propositional sarcasm-In Propositional sarcasm the scope is directed towards some proposition to which

\*Author for correspondence

sincere utterance have been committed by the speaker. Example-If you want a tasty food you should head over to KFC.

2. Like-prefixed sarcasm-Like-prefixed sarcasm provides denial of argument that is being made. Example-Oh, I talk with Ram all the time.
3. Illocutionary sarcasm-Illocutionary sarcasm involves non textual clues and indicates attitude opposites of that have been uttered.

Sarcasm detection is divided into three categories on the basis of text features that are being used for classification. The categories involve Lexical, pragmatic and hyperbolic feature based classification. Lexical feature based classification involves text properties such as uni-gram, bi-gram and n-grams. Pragmatic feature based classification refers to symbolic and figurative text. Examples-emoticons, smilies etc. Hyperbole feature based classification involves text properties such as intensifiers, interjections, punctuation mark, quotes etc.

In<sup>2</sup> explained how sarcasm can change the polarity of tweets from positive to negative. In<sup>3</sup> proposed rules to decide polarity of tweet when sarcasm is detected. They also showed that sentiment analysis performance could be enhanced when sarcasm is detected. In<sup>4</sup> viewed tools for sentiment analysis when sarcasm is detected. In this paper we have proposed way to detect sarcasm in tweets using Twitter API. In<sup>5</sup> viewed that lexical feature play important role for detection of irony and sarcasm in text. In<sup>6</sup> used the pragmatic features such as emoticons, smiles etc. for detection of irony in newspaper text data. In<sup>7</sup> viewed that tweets containing interjection words have higher chance of being sarcastic.

The tools used are Natural Language Processing Toolkit (NLTK). NLTK is set of programs for statistical natural language processing and is written in python language. NLTK consist of graphical representation and sample data. NLTK provides interfaces to WordNet with text processing libraries for classification, tokenization, stemming, tagging, parsing etc. Anaconda is used for processing of data. Anaconda is open source distribution of python programming language used for large scale processing of data. Anaconda consists of more than 720 open source packages and among these packages NLTK package is also installed already in anaconda. Anaconda comes with python pre-installed in it. The twitter data collected from Twitter API are then fetched to anaconda and then tokenization, stop word removal etc. are done

and then sentiment analysis is performed on the final data extracted. The polarity (positive or negative) of data is calculated and sarcasm is detected.

This paper consists of following steps:

1. Data is collected using Twitter Archiver. Data containing #sarcasm are detected.
2. Preprocessing of data is done. The Retweeted tweets and the tweets not containing English language are removed. English language tweets are selected using guess language package. Emojis are removed using Parse Charbase. Tokenization, Part of Speech Tagging and Stemming is done using TextBlob package installed in Natural Language Tool-Kit (NLTK). The upper cases letters are converted to lower cases.
3. Polarity of data is found that is whether data is positive, negative and neutral.
4. Data is trained and tested on Naïve Bayes and the accuracy is calculated using Weka.

For finding the polarity of data and their polarity confidence RapidMiner as well as TextBlob is used. RapidMiner is software that provides an environment for text mining, machine learning and analysis of data. RapidMiner supports the steps for machine learning such as data preparation, result visualization and validation. TextBlob is python library for processing textual data. TextBlob provides very simple API for various NLP tasks such as POS tagging, Parsing etc. Weka is used for finding accuracy of Naïve Bayes and SVM classifier. Weka stands for Waikato Environment for Knowledge Analysis

There are various limitations while working with tweets such as restriction of 140 characters only. The use of emojis, slangs used in internet and links to other images and websites. Thus the detection of which tweets should be considered as sarcastic and which tweets not to consider sarcastic is quite difficult.

Rest of the section contains related work, Preliminaries, Proposed approach, Result and discussion and finally conclusion.

## 2. Existing Methods

In the recent years researchers have provided more attention to sentiment analysis on twitter. Various papers have been addressed for sarcasm detection in twitter. In<sup>8</sup> provided a survey for automatic sarcasm detection. The paper provides various types of texts such as short text,

long text that are used in twitter. This paper also provides different approaches for sarcasm detection such as Rule-based Approaches, Statistical Approaches, Deep learning based approaches.

In<sup>9</sup> provided way for data collection using Twitter Archiver. This paper provides information about various packages which are used such as guess language package which is used for detecting tweets that only contain English language. Textblob packages which is used for part of speech in NLTK. This paper shows example of how Textblob is used in Part of Speech tagging. This paper also provides Classification Report of SVM and Naïve Bayes using the data set that is finally obtained by removing stopwords, emoticons, retweets etc. In<sup>4</sup> provided a behaviour modelling approach for sarcasm detection in twitter named as Sarcasm Classification using a Behavior Modelling Approach (SCUBA). According to SCUBA sarcasms have been categorized into various forms such as sarcasm detection as a contrast of sentiments, means of conveying emotions, form of written expression and complex form of expression.

In<sup>10</sup> provided various approaches for sarcasm detection in tweets such as Capturing and processing real time tweets using Flume and Hive, HMM based model for part of speech tagging and MapReduce functions for sarcasm detection. The MapReduce functions for sarcasm detection discussed in this paper are Parsing Based Lexicon Generation Algorithm (PBLGA) and Interjection Word Start algorithm (IWS). This paper has also viewed various other approaches for sarcasm detection in tweets such as Tweet contradicting universal facts, Tweets contradicting time dependent facts and Likes Dislikes Contradiction. This paper also shows framework for sarcasm analysis in real time tweets.

In<sup>11</sup> explained that sarcastic sentiment analysis can be divided into three categories such as lexical, pragmatic and hyperbolic feature based classification:

1. Lexical feature based classification- Lexical feature based classification involves text properties such as unigram, bigram and n-grams. Various authors have provided different paper for detecting sarcasm based on lexical feature based classification. In<sup>6</sup> explained for the first time how lexical features could be used for detecting sarcasm. In<sup>12</sup> viewed that bigram based features provide more better result for detecting sarcasm in tweets. In<sup>13</sup> explained various lexical features for detection of sarcasm. In<sup>10</sup> used bigram and trigram for generation of bag of lexicons to detect sentiment in tweets.
2. Pragmatic features based classification- Pragmatic feature based classification involves figurative and symbolic texts such as smilies, emoticons. Various authors have used pragmatic features for sarcasm detection. In<sup>7</sup> used pragmatic features such as emoticons and special punctuations mark to detect irony from newspaper text data. In<sup>13</sup> provided a sarcasm detection system using pragmatic features of data received from twitter. In<sup>14</sup> used pragmatic features for political tweets in order to predict the party which could win the election. In<sup>4</sup> used present and past tweets of users to detect their behavioral features and to detect sarcasm.
3. Hyperbole features based classification- Hyperbolic features based classification contains interjection, intensifiers, quotes and punctuation marks. Various authors have used hyperbolic features based classification to detect sarcasm. In<sup>15</sup> discussed adjectives and adverbs effect for intensification of text. In<sup>6</sup> provided detection of sarcasm using interjection and punctuation marks. In<sup>16</sup> viewed that phrase level and sentence level does not provide good accuracy and showed that text in document could improve accuracy. In<sup>8</sup> viewed that tweets having interjection words such as yay, wow etc. have more chance of being sarcastic. In<sup>17</sup> found concept level knowledge by using hyperbolic words in sentences. In<sup>10</sup> viewed the tweets that start with interjection words for detection of sarcasm.

Sarcasm and Irony are considered as same meaning words but they are different. Both sarcasm and irony refers to saying one thing and meaning another. The difference between sarcasm and irony is explained by<sup>18</sup>. In<sup>19</sup> explained role of ridicule in sarcasm and irony. In<sup>20</sup> explained contextual components used for conveying sarcastic irony in verbal form. In<sup>20</sup> also explained that sarcasm requires presence of four entities such as allusion for failed expectation, pragmatic insincerity, presence of victim and negative tension. In<sup>21</sup> told that sarcasm is not a discrete logical phenomenon unlike irony.

In<sup>22</sup> provided a method for detection of specific type of sarcasm that is where positive sentiment and negative sentiment contrasts. This paper provided the algorithm named bootstrapping algorithm which uses single word love and collection of tweets that are sarcastic to detect expressions showing positive statements and phrases

with respect to phrases showing negative situations. In<sup>28</sup> provided method for constructing a corpus of sarcastic tweets. They showed the impact of lexical and pragmatic factors on machine learning. In<sup>23</sup> used hashtag which people use in tweets to detect sarcasm. This paper provided a rule for deciding polarity of tweets depending on hashtags. In<sup>24</sup> used dataset with hashtag sarcasm. This paper provided approach for pattern extraction where words are used as content word list. The paper uses K-NN model for finding accuracy.

There are various papers that used Naïve Bayes and SVM for training and testing of data for finding the result. In<sup>25</sup> provided model for classification of tweets as positive and negative. In<sup>25</sup> developed sentiment classifier based on Naïve bayes method. In<sup>26</sup> used Naïve Bayes and Maximum entropy to classify tweets and found that Naïve Bayes performed better than Maximum entropy. In<sup>27</sup> extracted twitter data from Firehouse API and experimented on Naïve bayes, stochastic gradient descent and hoeffding tree and found that stochastic gradient decent provided better result when used with appropriate learning rate. In<sup>28</sup> viewed variations of Naïve Bayes classifiers in order to detect polarity of English tweets. Different variants of Naïve Bayes classifiers were build named as Baseline and Binary. Baseline is those where tweets are classified as positive, negative and neutral and binary are those which makes use of polarity lexicon and positive, negative and neutral are classified.

### 3. Preliminaries

This section describes Sarcasm Detection Engine, Removal of Stop Words, Tokenization, Part of Speech Tagging, Parsing and Sentiment Analysis.

#### 3.1 Sarcasm Detection Engine

The tweets are differentiated among positive, negative and neutral based on the detection engine. The tweets that are classified as positive, negative and neutral are further classified as Actual positive, Actual negative and Sarcastic sentiment. Actual positive tweets contain the tweets that are actually positive, Actual negative tweets are the tweets that are actually negative and Sarcastic tweets are those in which it is difficult to determine whether tweets are actually positive or negative. Sarcasm Detection Engine has been shown in the Figure 1.

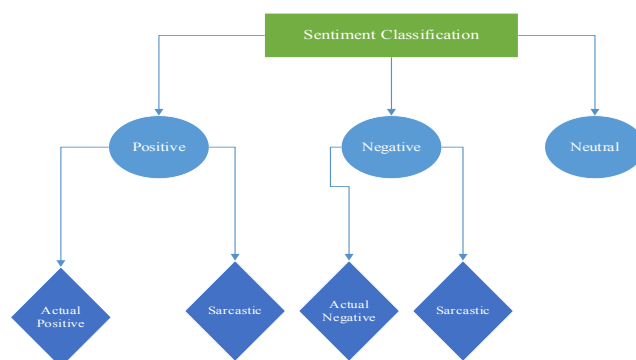


Figure 1. Sarcasm Detection Engine.

#### 3.2 Removal of Stop Words

The stop words are removed using python programming. The stop words corpus distributed with NLTK have been used. The stop words corpus consist of 2400 stop words for 11 different languages.

Example: “This is a long bar sentence”. After stop words removal we get ‘long’ ‘bar’ ‘sentence’.

#### 3.3 Tokenization

Tokenization is process of breaking texts into words, phrases and symbols called tokens. The tokens generated helps in parsing and performing other mining tasks. Tokenization is done using TextBlob package installed in NLTK.

Example: “Beautiful is better than ugly”. After tokenization using TextBlob we get every word separated such as ‘Beautiful’, ‘is’, ‘better’, ‘than’, ‘ugly’.

#### 3.4 Part of Speech Tagging

Part of Speech Tagging divides sentences into words and assign part of speech for each word. In this paper TextBlob is used for Part of Speech Tagging.

Example: “Sarcasm detection using twitter data”. Part of Speech tagging for the example is Sarcasm-NNP, detection-NN, using-VBG, twitter-NN, data-NNS. Here NNP stands for proper noun singular, NN stands for noun singular, VBG stands for verb present participle and NNS stands for noun plural.

#### 3.5 Parsing

Parsing is process of analyzing the symbols based on natural language. When a sentence is passed through parser, the parser divides the whole sentence into words and finds

Part of Speech (POS) tags. POS and syntactic relations are used to form units such as subject, verb and object. In this paper python based package named TextBlob is used for parsing.

Example: “Sarcasm detection using twitter data” is parsed giving result as Sarcasm/NN/B-NP/O, detection/NN/I-NP/O, using/VBG/BVP/O, twitter/NNP/B-NP/O, data/NNS/I-NP/O. Here O represents outside which means that the word is not part of a chunk.

## 4. Proposed Approach

In the proposed approach, the twitter data is collected using Twitter Archiver and the aim is to classify the sarcastic tweets as positive, negative and neutral. The aim is also to classify the tweets using Naïve Bayes classifier and SVM classifier and to differentiate between the accuracy, precision, recall and F-score of Naïve Bayes classifier and SVM classifier. A package named TextBlob that is installed in Natural Language Toolkit is used for preprocessing of the twitter data. The preprocessing steps involve tokenization, part of speech tagging and parsing. The stop words are removed using python programming. The stop words corpus which consist of 2400 stop words and which is distributed with NLTK have been used. RapidMiner, a tool have been used for finding the polarity and subjectivity of the data. TextBlob is used to find the polarity and subjectivity confidence of the data. Weka has been used to find the accuracy, precision, recall and F-score using Naïve Bayes classifier and SVM classifier. The results are compared between Naïve Bayes and SVM classifier. Overview of proposed approach has been shown in the Figure 2.

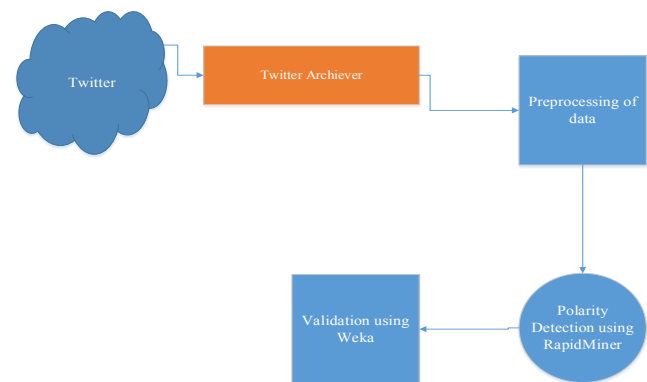
### 4.1 Working of Proposed Approach

#### 4.1.1 Data Collection using Twitter Archiver

The data in this paper is collected using Twitter Archiver. Twitter Archiver helps to save tweets for any search keyword or hashtags in the Google Spreadsheet. With help of Twitter Archiver the data containing hashtag sarcasm is collected. Data of 650K is collected with hashtag sarcasm. The data used from Twitter Archiver contains Full name of person tweeted, Tweet text, Tweet ID, the type of system by which he had uploaded the data, the number of followers the user have, the number of retweets, location of the user and whether user is verified or not. The data was also collected using Twitter API which constituted

data of year 2009. Observing the real tweets and the data in csv format it was observed that some HTML markup were present in data such as < and > . These were converted into their original form. The slangs that are used in twitter are also converted into their original form. Some of the mostly used slangs and their converted forms are as follows:

- a) “b/c” is converted to “because”
- b) “BRB” is converted to “Be Right Back”
- c) “BTW” is converted to “By The Way”
- d) “LOL” is converted to “Laugh out Loud”
- e) “FYI” is converted to “For Your Information”
- f) “zzz” is converted to “Sleeping”
- g) “TTYL” is converted to “Talk to you later”
- h) “ASAP” is converted to “As Soon As Possible”
- i) “ASL” is converted to “Tell me your Age, Sex and Location”



**Figure 2.** Steps involved in Proposed Approach.

#### 4.1.2 Polarity Detection Using RapidMiner

For polarity detection RapidMiner is used. RapidMiner works with Drag and Drop. The operators are dragged from operator section and dropped in process section. Polarity Detection using RapidMiner involves certain steps which are as follows:

1. The operator named “Read Excel” is dragged from operator section and dropped to process section. In the “ReadExcel” operator the excel file which contains the data is selected. The twitter data has been collected using Twitter Archiver.
2. The analysis of sentiment takes place. In order to clean the data and remove unwanted terms operator named “Process Document from Data” is used. The “Process Document From Data” contains various



operators within such as Transform Cases, Tokenize, Filter Stopwords, Filter Token by Length and Stem (Porter) operator. The “Transform Cases” operator is set to lower case so that all the upper case letters are converted to lower cases. The “Tokenize” operator is set to non letter mode. This is done so that whenever space or full stop is found, the word will be split over there that is the string will be split into words over there. The “Filter Stopwords” operator is used to filter the stopwords from the data. “Filter Token by Words” operator is used to filter tokens by length. The limit of minimum and maximum characters could be set here. The Stem Porter operator is used for stemming of words such as taken is stemmed to take. The “Process Document from Data” is set to TF-IDF. TF stands for Term Frequency and IDF stands for Inverse Document Frequency. Term Frequency refers to number of times a term occurs in a document. Inverse Document Frequency reduces the weight of terms that occur very frequently in the document and increases the weight of terms that occur rarely.

3. “Set Role” operator is used. In the “Set Role” operator attribute name is set to Sentiment so that the sentiment of text could be found out. The target role is defined as Label. The output of “Process Document from Data” operator acts as input to the “Set Role” operators.
4. “Select Attribute” operator is used. The “Select Attribute” operator is used to filter out any attributes that are having missing values. The filter type set in select attribute operator is no missing values. The no missing value helps to filter out missing values.
5. “Split Validation” operator is used. The Split ratio in “Split Validation” operator is set to value of 0.66. Setting the Split ratio to 0.66 means that two-third part of the data will be used for training and the remaining one-third of the data is used for testing. Within the validation operator Training and Testing phase are set.

In this paper we have used Naïve Bayes for training the data. The training data is given input to operator and the operator will generate the model. The “Apply Model” operator is used to apply the model in testing the dataset and the “Performance Operator” is used to measure the performance of the model.

#### 4.1.3 Validation Using Weka

Weka is used for finding accuracy of sarcastic data using Naive Bayes Classifier and SVM classifier. For calculation of accuracy a total of 2250 tweets were used. The classifications of tweets as positive, negative and neutral are done using RapidMiner. The steps involved in validation are as follows:

1. The twitter data is converted into weka file format of arff by using the following command.
2. The tweets are classified using SVM Classifier and Naïve Bayes Classifier and the correctly classified instances of the data is found.

## 5. Result and Discussion

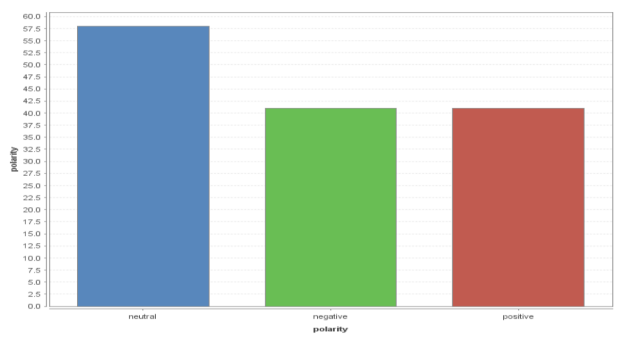
The result obtained using RapidMiner contains few samples of tweets and their polarity that is whether the tweets are positive or negative. The result obtained also contains the polarity confidence. Confidence shows the accuracy of polarity. Subjectivity refers to interpretation based on personal feelings, emotions etc. Objectivity is not influenced by emotions, personal feelings and opinions. The results obtained are shown in the following Table 1.

Like-wise the Table 1 a sample of 140 tweets has been collected and the tweets have been analyzed on basis of polarity, subjectivity, polarity confidence, and subjectivity confidence using graphical form. There are four graphs showing graphical structure of polarity, subjectivity, polarity confidence, subjectivity confidence.

**Table 1.** Polarity and Subjectivity of tweets along with their confidence

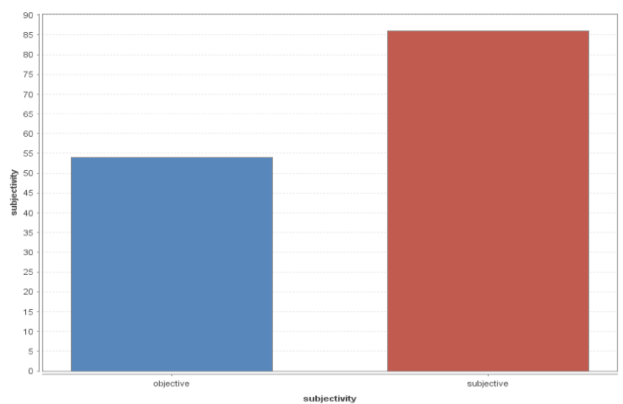
Row No.	Tweets	Polarity	Polarity Confidence	Subjectivity	Subjectivity Confidence
1.	People misperceiving Trump infomercials	neutral	0.400	Objective	1.000
2.	Sometimes tiring deal with people	negative	0.628	Subjective	1
3.	Donot know sounded like locker room talk	neutral	0.580	Subjective	1
4.	Offended hate speech against bigfoot	negative	0.976	Objective	1.000
5.	Should go great cause no problem	positive	0.605	Objective	1.000
6.	Choosing colonialist storyline groundbreaking	positive	0.674	Subjective	1.000

Figure 3 shown provides the classification on basis of polarity of tweets that is the dataset being used contains what number of positive, negative and neutral tweets. As from the graph the neutral tweets more than positive and negative tweets. From 140 tweets 41 tweets are positive, 41 tweets are negative and 58 tweets are neutral. The fractions of positive and negative tweets are 0.293 and the fractions of neutral tweets are 0.414.



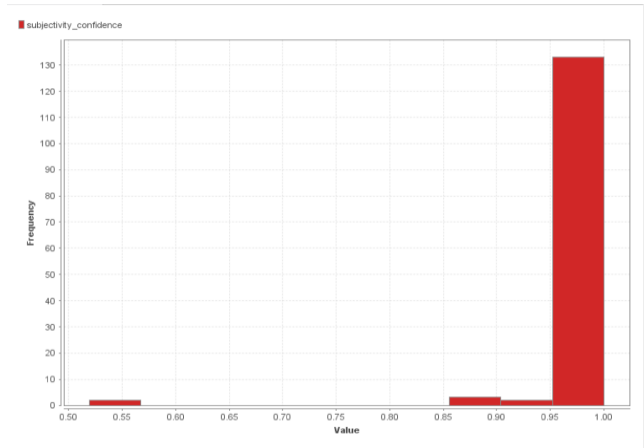
**Figure 3.** Classification on basis of polarity of different tweets.

Figure 4 shows the classification of tweets on basis of subjectivity. The tweets are classified as objective and subjective. Subjective is based on personal feelings and emotions whereas objective is not influenced by personal feelings and emotions. From 140 tweets 86 are subjective and 54 are objective. The fraction of subjective is 0.614 and the fraction of objective is 0.386.



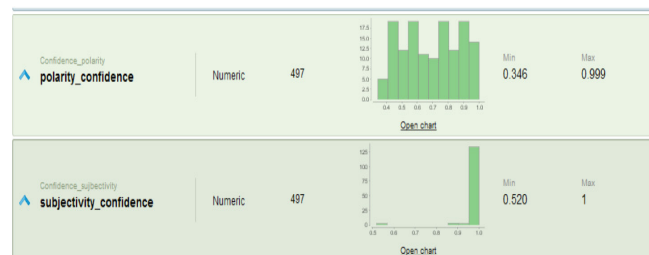
**Figure 4.** Classification of tweets on basis of subjectivity.

Figure 5 shows the subjectivity confidence of tweets. The graph shows that most of the tweets are fully subjective with value ranging from 0.96 to 1.



**Figure 5.** Subjectivity confidence.

Figure 6 shown above provides Min and Max value of polarity and subjectivity confidence along with their graph. The minimum value of polarity confidence is 0.346 and maximum value is 0.999. The average of polarity confidence is 0.691 whereas the standard deviation is 0.184. The minimum value of subjectivity confidence is 0.520 and maximum value is 1. The average of subjectivity confidence is 0.989 whereas the standard deviation is 0.058.



**Figure 6.** Min and Max value of polarity and subjectivity confidence along with their graphs.

Weka is used for determining correctly detected instances using SVM classifier as well as Naïve Bayes classifier and also to determine Precision, Recall and F-score of SVM and Naïve Bayes classifier for twitter data shown in the Table 2 result is based on using 2250 tweets.

**Table 2.** Validation using Naïve Bayes and SVM classifier

	Naïve Bayes	SVM
Accuracy	65.2%	60.1%
Precision	96.5%	98.2%
Recall	20.1%	20.4%
F-score	37.4%	33.7%

Precision is total number of correct results divided by the number of all returned results. Precision measures the number of tweets that have successfully been classified as sarcastic over total number of tweets that are classified as sarcastic.

Recall is number of correct results divided by number of results that should have been returned.

F-score considers both precision and recall to calculate the score. Formula for F-score is  $2 * [(precision * recall) / (precision + recall)]$ .

## 6. Conclusion

Sarcasm detection and analysis of twitter provides opinion about public opinion on the recent trend and events. This paper found the polarity of tweets using RapidMiner. Various graphs have been shown of polarity, subjectivity, subjectivity confidence and polarity confidence. A total of 2250 tweets were used for calculating the accuracy of SVM classifier and Naïve Bayes classifier. The accuracy of Naïve Bayes is 65.2% and of SVM is 60.1%. Thus Naïve Bayes provides more accuracy as compared to SVM classifier.

## 7. References

1. Cambell JD, Katz AN. Are there necessary conditions for inducing a sense of sarcastic irony, *Discourse Processes*. 2012; 49 (6):459-80. Crossref.
2. Liebrecht C, Kunneman F, Bosch AVD. The perfect solution for detecting sarcasm in tweets# not. Radbound University. 2013; 1-9.
3. Maynard D, Greenwood MA. Who cares about Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis. 2014; 1-6. PMid:25320772.
4. Rajadesingan A, Zafarani R, Liu H. Sarcasm detection on twitter: A behavioral modeling approach, *Web Search and Data Mining*. 2015; 97-106. Crossref.
5. Ravichandran SP. Sarcasm Detection in Twitter data, *Rochester Institute of Technology*, 2015; 1-26. PMid:25450559.
6. Kreuz RJ, Glucksberg S. How to be sarcastic: The echoic reminder theory of verbal irony, *Journal of Experimental Psychology*. 1989; 118(4):374-86. Crossref.
7. Carvalho P, Sarmiento L, Silva MJ. Clues for detecting irony in user-generated contents: oh!! It's so easy. *Proceedings of the 1st*. 2009; 53-56. Crossref.
8. Lunando E, Purwarianti A. Indonesian social media sentiment analysis with sarcasm detection, *Advanced Computer Science*. 2013; 195-98. Crossref.
9. Joshi A, Bhattacharyya P, Carman MJ. Automatic sarcasm detection: A survey. *arXiv preprint arXiv:1602.03426*. 2016.
10. Bharti SK, Vachha B, Pradhan RK, Babu KS. Sarcastic sentiment detection in tweets streamed in real time: A big data approach, *Digital Communications*. 2016; 1-17. PMid:PMC4741291.
11. Pradhan VM, Vala J, Balani P. A Survey on sentiment analysis algorithms for opinion mining, *International Journal of Computer*. 2016; 133(9):1-5.
12. Tsur O, Davidov D, Aappoport A. ICWSM-A great catchy name: semi-supervised recognition of sarcastic sentences in online product reviews. 2010; 1-8.
13. Tayal DK, Yadav S, Gupta K, Rajput B. Polarity detection of sarcastic political tweets, *Sustainable Global*. 2014. Crossref.
14. Utsumi A. Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from non-irony, *Journal of Pragmatics*. 2000; 32(12):1777-806. Crossref.
15. Filatova E. Irony and sarcasm: Corpus generation and analysis using crowd sourcing, *LREC*. 2012; 1-7.
16. Tungthamthiti P, Shirai K, Mohd M. Recognition of sarcasms in tweets based on concept level sentiment analysis and supervised learning approaches, *PACLIC*. 2014; 1-10.
17. Giora R. On irony and negation. *Discourse processes*, Taylor and Francis. 1995; 19(2):239-64.
18. Lee CJ, Katz AN. The differential role of ridicule in sarcasm and irony, *Taylor and Francis*. 1998; 13(1):1-15.
19. Toplak M, Katz AN. On the uses of sarcastic irony, *Journal of Pragmatics*. 2000; 32(10):1467-88. Crossref.
20. Nakamura SK, Glucksberg S. How about another piece of pie: The allusional pretense theory of discourse irony, *Journal of Experimental*. 1995; 124(1):3-21.
21. Riloff E, Qadir A, Surve P, Silva DL, Gilbert N. Sarcasm as contrast between a positive sentiment and negative situation. 2013; 1-11.
22. Maynard D, Greenwood MA. Who cares about sarcastic tweets? Investigating the Impact of Sarcasm on Sentiment Analysis, *LREC*. 2014; 1-6. PMid:25320772.
23. Davidov D, Tsur O. Semi-supervised recognition of sarcastic sentences in twitter and amazon, *ACM*. 2010; 1-10.
24. Pak A, Paroubek P. Twitter as a corpus for sentiment analysis and opinion mining, *LREC*. 2010; 1-7.
25. Parikh R., Movassate M. Sentiment analysis of user-generated twitter updates using various classification techniques, *CS224N Final Report*. 2009; 1-18.
26. Bifet A, Frank E. Sentiment knowledge discovery in twitter streaming data, *International Conference on Discovery Science*. 2010; 1-15. Crossref.
27. Gamallo P, Garcia M. Citius: A naive-bayes strategy for sentiment analysis on English tweets, *Proceedings of SemEval*. 2014; 1-5.
28. Ibáñez RG, Muresan S. Identifying sarcasm in Twitter: A closer look, *ACM*. 2011; 581-86.