# SARCASM DETECTION FOR ENGLISH TEXT

*Riya Das, Shailey Kadam, Chetan Kalra, Vijeta Nayak and Dr. Sharvari Govilkar*
*Department of Computer Engineering, Mumbai University, PCE, New Panvel, India*

***Abstract— Sarcasm determines the mockery or irony used by that person to express his emotions. With the increase in the use of social medias which is mostly in the form of text, it becomes important to detect the sarcasm present in the sentences. So understanding the sentiments of the text becomes very important. In our previous paper* [14] *we proposed a conceptual framework for Sarcasm detection using three machine learning algorithms Viz. Random forest, Naive Bayes, SVM. Our training consists of Twitter dataset with emoticons, punctuations, hashtags and other dataset from different sites. This paper describes the processing steps and the actual workflow and compares the best algorithm among the three algorithms for future work purposes.***

***Index Terms* — Hashtags, Punctuation Marks, Emoticons, Random Forest Classifier, SVM, Naive Bayes Classifier**

## 1. INTRODUCTION

Our objective is to use the concept of machine learning in order to train and test various sentences. Hence, this paper presents a method for detecting sarcasm in given text. Our dataset is a collection of tweets and various reviews with 46,000+ sentences.

Since our project mainly focuses on English text, the most important process is to remove all other mixed languages present in the given statement. This is done by script validation and filtering of pre-processing block. Before training any dataset first step is to clean the noise present in the dataset, which is done by preprocessor block by removing stop words and HTML tags. This cleaned data is then used to train classifiers such as Random Forest, SVM and Naive Bayes. The dataset is divided approximately into 70-30% in order to train and test data to get the desired result. A confusion matrix is then formed which helps us to understand the number of false positives and false negatives during the training part. This paper also deals with comparing these result to find out which classifier gives a better result and accuracy so that the best classifier can be used in social media analytics in order to improve the overall sentiment of these statements. The scope of the system would be to find the Sarcasm present in English Language Only.

The recipients of the system would be organizations which use social media monitoring such as public opinion, reviews and rating of the product which provide valuable information about emerging trends and what consumers and clients think about specific topics, brands or products.and also with the rapid development of craze TV series, use of sarcasm in daily life has become more common and prominent. Besides this, use of Hashtags and emoticons have rapidly been increasing. Therefore, it has become a need of an hour for all these companies to understand the progress of their products in the market and among their clients.

## 2. LITERATURE SURVEY

As discussed in our previous paper [14] we can conclude that though sarcasm can be determined with a lexicon based approach, but it would take more time for computation. While if we can obtain the features and store it in a file, we can reuse the same featured for determining sarcasm any number of times without actually performing all the processes. Therefore, our project mainly focuses on supervised machine learning approach as it is better to train and store the features, and use them for testing other sentences.

## 3. SARCASM DETECTOR

In this, we would be discussing about the system architecture. The input of the system would be reviews or simply some content from various Social Media Sites and tweets from twitter, etc.. The first step is to clean the raw input so that a standardized format of content is obtained. From the cleaned data, we have constructed our dataset which is used in training phase to train the various machine learning classifiers.

Few preprocessing of data is done like script validation, removal of URLs and HTML tags. This cleaned data is then converted into standard format i.e data matrix with reviews and labels. labels is of two types 0 and 1 indicating the sentence being not sarcastic and sarcastic respectively.

Training data consists of hashtags, emoticons, punctuation marks and too positive and negative sentences, therefore there is no need to handle them separately. The system uses three supervised machine learning algorithm, such as **Random Forest**, **Support Vector Machine (SVM)** and **Naive Bayes Classifier** to train and test the dataset.

In training phase the algorithm builds a classifier by analysing the training data and associated label with each class and creates a pickle file which consists of all the features extracted by the model in the training phase. From the data model created, a confusion matrix is generated which help us to find the number of true positives, true negatives, false positives and false negatives during the training phase to understand how accurately the data is being trained by each classifier. During the testing phase, the system accepts the input from the user and compares with the features stored in the pickle file and predicts whether the given input sentence is sarcastic or not.

The main aim of the system is to compare these algorithms to find which algorithm can be further used to detect sarcasm during text analytics.

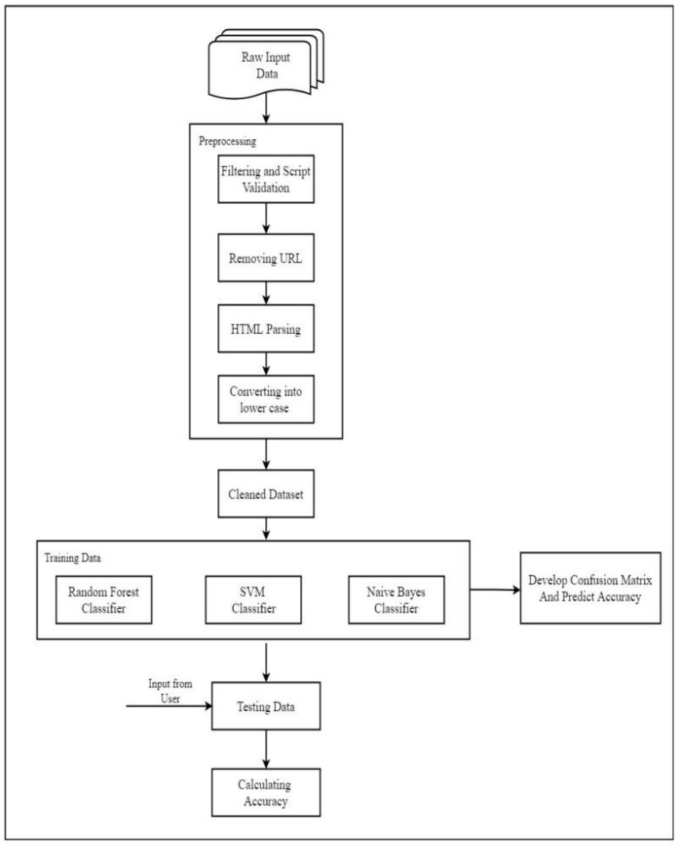

Figure 1 : Sarcasm Detector

*3.1 Input Documents*

The text will be in Romanized English format. The content would be collected from different social media domains like Twitter or from product based websites like Amazon, etc.

*3.2 Preprocessing Block*

The process of converting raw input data collected from various social media sites and twitter into standardized format of data matrix i.e label and review.

*3.2.1 Filtering and Script Validation*

The process of considering only English text by ignoring all the mixed language text so that processing of text can be made easier.

In this step, the given sentence is scanned character by character and compared with UTF-8. If character is present in the given list, then it does not belong to English Script and hence can be ignored.

*3.2.2 Removing URLs*

The process of removing all unwanted text such as URL so that more informative data can be stored in the dataset for training.

Algorithm :

```
a.  Input : The sentences only containing English Text
    and special characters like hashtags, emojis,
    punctuation marks, etc.
b.  Output : URL present in the sentence are removed.
c.  Steps :
    i.   START.
    ii.  Define a regular expression to identify the
         presence of https://www.abc.com
    iii. Scan the input document.
    iv.  Check for not End of file.
         1. Read a character from input file.
         2. IF character matches with regular
            expression then remove it.
         3. Display the text after removing text
            otherwise go to step 4.
         4. Read the next input sentence.
         5. STOP.
```

*3.2.3 Removing HTML Tags*

The process of removing all unwanted text such as HTML tags so that more informative data can be stored in the dataset for training.

Algorithm :

```
a.  Input :  Sentences with no URLS.
b.  Output : Sentences without any HTML tags.
c.  Steps :
    i.   START.
    ii.  Identify all predefined HTML Tags by using
         predefined packages.
    iii. If the sentences contain any html Tags then
         remove it and display it otherwise go to next
         step.
    iv.  Read the next input sentence.
    v.   Presence of HTML tags can be compared by
         comparing the input and output string of this
         block.
    vi.  Repeat the same process until end of document
         is found.
    vii. STOP.
```

*3.2.4 Converting into Lower Case*

This block converts the input string into one standard format which is in lower case.

*3.2.5 Clean Dataset*

This block contains dataset free from all unwanted URL, HTML tags and converted into Lower Case.
Stop words are not removed during pre processing as it might contain some sentiments that would affect its meaning. In this blocks labels are assigned to each sentences and are stored into standardized format i.e review and its corresponding label. Labels are in form of 1 and 0 which represent sentences are sarcastic or non - sarcastic respectively.

*3.3 Training Classifier*

Data Classification is termed as the process that organizes data into categories so that it can be used efficiently and effectively. It basically has two phases :

a. **Training Phase :** At this phase, the classification algorithm uses the training data for analysing.
b. **Testing Phase :** In this phase, testing data are used to estimate the accuracy of the classifier. Testing data is the dataset used for evaluating the model in the training phase.

Based upon the data chunk the dataset is divided for training and testing. Ideally we used 70-30% to train and test data respectively.

*3.3.1 Tf-idf*

The TF (term frequency) of a word is the frequency of a word (i.e. number of times it appears) in a document.
The IDF (inverse document frequency) of a word is the measure of its importance in the whole corpus.
The formula for to measure Tf-idf is :

$tfidf(t,d,D)=tf(t,d)*idf(t,D)$……………………………(3.1)

Where t denotes the terms; d denotes each document; D denotes the collection of documents.

*3.3.2 Random Forest Classifier*

Random forest algorithm is one of the supervised learning classification algorithm. This classifier generates large number of decision trees and randomly selects the best node from which features can be extracted and stored.
With increased number of trees for predication will automatically gives higher accuracy results. Hence, of our system we have generated maximum number of trees which help us to extract features for the classifier.

Algorithm for Random Forest can be divided into two phases :
i. Train the Dataset
ii. Random Forest Prediction

Algorithm :

i. Define parameters using TfidfVectorizer.
ii. Train the classifier with the parameters defined.
iii. Make predictions of data from training dataset.
iv. Find accuracy and confusion matrix for training and testing dataset.
v. Plot confusion matrix.

*3.3.3 Support Vector Machine*

A Support Vector Machine (SVM) is also one of the supervised machine learning algorithm that can be used for both classification and regression purposes. It is mainly used in classification problems.

In this algorithm, each data item is plotted against hyper plane in space with its feature extracted as the value od data item. The data points which are nearest to the defined hyper plane is called as support vectors.

i. CountVectorizer : It converts a collection of text documents to a matrix of token counts. This implementation produces a sparse representation of the counts.
ii. SGDClassifier : SGD stands for Stochastic Gradient Descent where the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule.
iii. GridSearchCV : If it is not used we need to loop the parameters and run all the combination of parameters. For this we need to write the code manually which increases the time requirements.

Hence, for our system we have used GridSearchCV.

Algorithm :

i. Defining various parameters using SGDClassifier.
ii. Use GridSearchCV to iterate the parameters automatically.
iii. Train the classifier based upon parameters defined.
iv. Make predictions of data from training dataset.
v. Find accuracy and confusion matrix for training and testing dataset.
vi. Plot confusion matrix.

*3.3.4 Naive Bayes Classifier*

Naive Bayes Classifier is based on the Bayesian theorem. It is suitable where the dimensionality of the input attributes is high. In this model, parameter estimation is done by using maximum likelihood. It is used to find conditional probabilities.

P(X|Y) is the conditional probability of event X occurring for the event Y which has already been occurred.

$P(X|Y)=P(X \text{ and } Y)/P(Y)$……………………………….(3.2)

a. MutinomialNB Classifier : For our system we have implemented MultinomailNB which makes use of the Naive Bayes algorithm for multinomially distributed data. The parameters is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting.

Algorithm :
i. Define parameters using TfidfVectorizer and MultinomialNB.
ii. Training the classifier with the parameters defined.
iii. Make predictions of data from training dataset.
iv. Find accuracy and confusion matrix for training and testing dataset.
v. Plot confusion matrix.

## 4. Result Analysis

Training dataset is generated by cleaning the raw data collected from various social media sites like Amazon, Facebook, etc. and tweets from twitter. For the evaluation of our system, we have used 10,000 sentences of each type for each classifier model. The system extracts the features from the input sentence and compare it with the features stored in pickle file to detect whether the given input sentence is sarcastic or not.

Example 1 : Apparently I was not supposed to be happy :unamused_ face:
Random Forest : Yes
SVM : Yes
Naive Bayes Classifier : Yes
Expected Outcome : Sarcastic

Example 2 : I am going to take a leave from office today.
Random Forest : No
SVM : No
Naive Bayes Classifier : No
Expected Outcome : Non - Sarcastic

Example 3 : Whatever it is that is eating you, it must be suffering horribly.
Random Forest : No
SVM : No
Naive Bayes Classifier : No
Expected Outcome : Sarcastic

The efficiency of our system is based on the confusion matrix generated after training the classifier and number of correct output given by each classifier for input sentence during testing.

The following graph shows the accuracy obtained by the system during training phase.

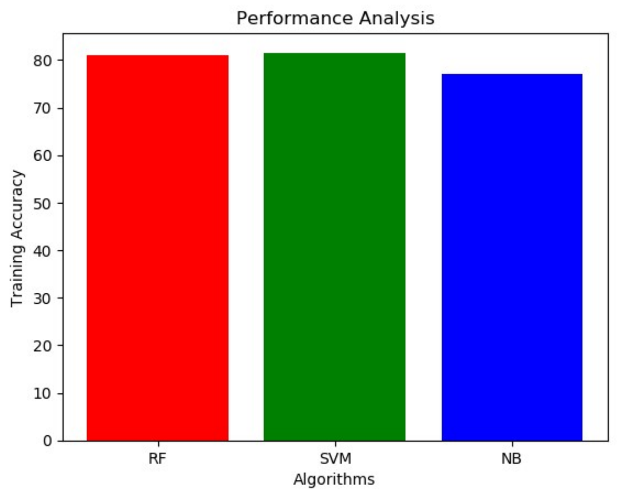

Figure 2 : Performance Analysis for Training Phase

Therefore the graph below helps us to compare which algorithm is best to classify the sentences into sarcastic and non-sarcastic respectively.

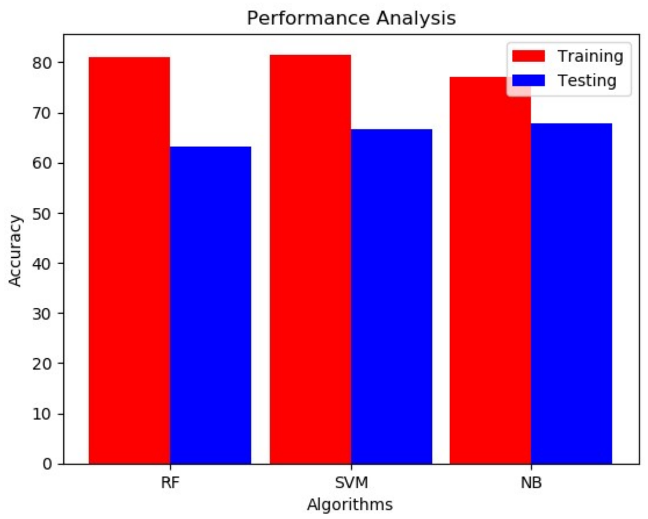

Figure 3 : Accuracy Comparison

## 5. Conclusion

Every algorithm has its own advantages and completely different process to identify patterns. The training accuracy obtained after training the three classifier is as :

Table 1 : Training Accuracy

| **Algorithms** | **Accuracy** |
|---|---|
| Random Forest | 81% |
| SVM | 81.54% |
| Naive Bayes | 76.99% |

While testing accuracy obtained after evaluating 10,000 dataset of each is as :

Table 2 : Testing Accuracy

| **Algorithms** | **Accuracy** |
|---|---|
| Random Forest | 63.09% |
| SVM | 66.74% |
| Naive Bayes | 67.81% |

The Naive Bayes algorithm performed better than the other two algorithm performed for identifying similarities between non sarcastic and sarcastic sentences respectively whereas by using Support Vector Machine, system has a slight edge for extracting sarcastic patterns.

Our System compares the best machine learning algorithm from the three algorithms viz. Naive Bayes, Random Forest and SVM to detect sarcasm present in the given text. It gives us the desired output from the features obtained during the training phase. But due to false positives and false negatives obtained while training, sometimes, this system predicts a wrong output. But this can be further improved by using deep learning techniques like Keras and Tensor-flow. Classifiers can be made more powerful by training more amount of dataset with emoticons which might increase the accuracy of the classifier.

Acknowledgment

We would like to thank Dr. Madhumita Chatterjee, Head of Computer Engineering Department for the invaluable support. We would also like to show our gratitude towards Dr. Sandeep M. Joshi, Principal, PCE, New Panvel for his invaluable support and for providing an outstanding academic environment. Last but not the least we would like to thank Prof. Dhiraj Amin and all other staff members of the Department of Computer Engineering for their critical advice and guidance without which this project would not have been possible. Also, we would like to say that it has indeed been a fulfilling experience for working out this project topic.

## REFERENCES

[1] Whiting A and D Williams. Why people use social media: a uses and gratications approach. Qualitative Market Research: An International Journal, 2013

[2] Ilia Vovsha Owen Rambow Apoorv Agarwal, Boyi Xie and Rebecca Passonneau. Sentiment analysis of twitter data. In Proceedings of the ACL 2011 Workshop on Languages in Social Media, pages 30-38, 2011.

[3] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In Proceedings of COLING, pages 36-44, 2010.

[4] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. 2010.

[5] Bhyani R. Go, A. and L Huang. Twitter sentiment classification using distant supervision. Technical report, CS224N Project Report, Stanford, 2009.

[6] Daniel Neagu Haruna Isah, Paul Trundle. Social media analysis for product safety using text mining and sentiment analysis. 2015.

[7] Yulan He Hassan Saif and Harith Alani. Semantic sentiment analysis of twitter. 2011.

[8] P. Anderson J. Blackburn C. Borcea N. Kourtellis, J. Finnis and A. Iamnitchi. Prometheus user-controlled p2p social data management for socially aware applications. 2010.

[9] B. Pang and L. Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2008.

[10] Ashwin Rajadesingan, Reza Zafarani Arizona, and Huan Liu Arizona. Sarcasm detection on twitter: A behavioral modeling approach. 2015.

[11] Hiroshi Shimodaira. Text classification using naive bayes. 2015.

[12] https://github.com/AniSkywalker: -Dataset

[13] http://scikit-learn.org/stable/

[14] Conceptual Framework For Sarcasm Detection for English Text - Riya Das, Shailey Kadam, Chetan Kalra and Vijeta Nayak (Department of Computer Engineering, Mumbai University, PCE, New Panvel, India).