# Sentimental Analysis of Twitter Dataset Using Classification Techniques

Sarthak Gupte

***Abstract* - Semantic Analysis is used a lot on the Twitter Dataset, to draw exact meaning and to classify similar types of tweets together. In this paper we will create a Twitter Dataset and will apply different NLP techniques to pre-process the data. The pre-processed data is provided to different classification models. Later on we will be comparing different ML models on the basis of the result obtained, to rank them in the order of the best performance. Here we are only using the classification technique to predict the class, as we will only classify the tweets into three underlying groups, Positive and Negative.**

***Keywords* - NLP -Natural Language Processing, ML - Machine Learning**

## 1. INTRODUCTION

Sentimental analysis is a mixture of Natural Language Processing(NLP) technique and Machine Learning process, used to identify the real meaning of the given text by the machine. It helps the machine to classify a given text into an emotion, generally from positive and negative. The categories may vary depending upon the requirements of the machine, but the basic categories are positive and negative. Sentimental Analysis is used to obtain quick perception about the large texts this helps save time and to gain insight before even reading the text. Big corporations and companies uses Sentimental Analysis to understand the response for their product with actually reading every review or comment written by the users.

Twitter is a social networking service used by almost 330 million users and 145 million daily users to write post, share information, picture and other media. Posts shared on the Twitter is called tweets. Since twitter has a large amount of users using their services to post tweets daily, they have a huge database which consists of information regarding most of the trending topics, major products in the market. Basically, twitter is very rich in data with very diverse topics. This might help anyone to review a product with the use of sentimental analysis. Majority of the customer by the product after reading the reviews from social media platform. Sentimental Analysis will help to make this review process lightning quick, as there is no need to read a review or post to know whether it is a positive or negative comment.

In this project, we will be performing a Sentimental Analysis on the Twitter dataset. Dataset will be scrapped through the Twitter API using python language. Aim of this project is to classify the emotion of the tweet into either positive or negative.

Although the Twitter dataset is rich in data, it is difficult for analysis because of the word limit, emoticons, abbreviations, hashtag and etc. To get the data on which we can apply ML models, firstly we will pre-process the data to remove stop words and other unwanted data. After obtaining the cleaned dataset we will be using the ML models to classify the tweets into our known underlining groups.

The timeline of this project will be as follows. First, getting data from the Twitter API using scrapping methods. Second, pre-process the data. Third, performing NLP and ML techniques on the data to classify the data into the groups. Fourth compare and contrast the different classification techniques used for the sentimental analysis.

## 2. MOTIVATION

It is difficult to predict an overall reaction of the users on any social media platform for a given topic. In order to identify whether a topic have a negative or a positive response from majority of the users, we could use semantic analysis. The main motivation for writing this paper is to understand and evaluate when a topic is widely accepted by the people and when a topic is not accepted.

## 3. RELATED WORK

Many researchers and analyst has already worked under this area. In this section we will be discussing three related works.

Ramadhani et al. [3] proposed a method to analysis sentiments of the given tweets in two language Korean and English. Firstly, the authors perform text mining and text cleaning then use the ML model for sentimental analysis which includes stemming and removal of stop words. In this model the authors used a multi layer perceptron, which is a single layer neural network.

Neethu et al.[2] talks about sentimental analysis of twitter dataset. Firstly, the author pre-process the data by removing slangs and spelling mistakes and remove urls. Then for classification vector is created from the data which is further

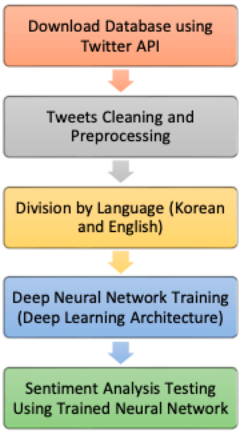
Download Database using Twitter API
Tweets Cleaning and Preprocessing
Division by Language (Korean and English)
Deep Neural Network Training (Deep Learning Architecture)
Sentiment Analysis Testing Using Trained Neural Network

**Figure 1: Flow of Sentimental Analysis[3]**

used to classify using Naive bayes, SVM, maximum entropy and ensemble classifier. Further they compare the results obtained from different classifiers.

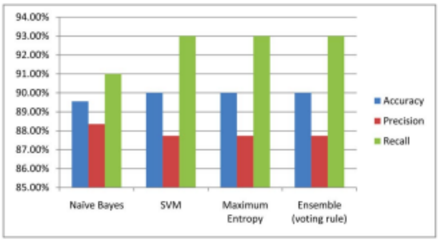

**Figure 2: Comparisons of the classifiers[2]**

Kanakara et al.[1] compared 9 ML models consisting of supervised and unsupervised models on the pre-processed data created using removal of multiple words, lemmatization and removal of stop words. The system created uses specific modules. Data gathering modules, data processing modules, training and classification module and finally classification outputs.

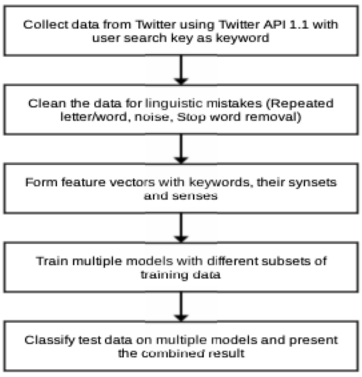
Collect data from Twitter using Twitter API 1.1 with user search key as keyword
Clean the data for linguistic mistakes (Repeated letter/word, noise, Stop word removal)
Form feature vectors with keywords, their synsets and senses
Train multiple models with different subsets of training data
Classify test data on multiple models and present the combined result

**Figure 3: Proposed System[1]**

## 4. DATA DESCRIPTION

Twitter is a social media platform which allows users to post, share, comment, reply etc. The messages that an user can post on Twitter is known as ”tweets”. Tweets can contain different type of emoticons, hashtags, and could be written in different languages. Tweets are short messages with the word limit of 140 words.

To scrap data from twitter a python library named Tweepy is used. Which connects with Twitter API and then officially using Tweepy's method we can scrap data by applying multiple filters on hashtag, geo location etc. To access this API you need to enable developer mode with your Twitter account. After enabling some unique keys would be generated by Twitter which should not be shared with anyone. Tweepy allows us to limit the number of tweets that we can scrap, if you want to scrap 1000 tweets it will take roughly 2-3 minutess. But if we want scrap tweets up to 100000, it will take approxximately 15-20 minutes. The data obtained using tweepy could be repeated. To remove duplicate records from the data we will be using some pre-processing techniques later on.

The dataset created using the described python library, consists of following attributes.

| Attribute | Description |
|---|---|
| screen_name | Alias name used on screen. |
| username | Name of the user. |
| user_id | Unique ID of user. |
| tweet_id | Unique Id of the tweet. |
| tweet_url | URL of the tweet. |
| timestamp | Time when the tweet was posted. |
| timestamp_epochs | |
| text | Text written in the tweet. |
| text_html | Text written in HTML format. |
| links | Link of the tweet. |
| hashtags | Hashtags used in the tweet. |
| has_media | True if media is there else false. |
| img_urls | If image present then URL. |
| video_url | If video present then URL. |
| likes | Likes on the tweet. |
| retweets | Retweets on the tweet. |
| replies | Replies on the tweet. |
| is_replied | |
| is_reply_to | . |
| parent_tweet_id | parent id of the tweet. |
| reply_to_users | replied to following |

Since, the keys cannot be shared for this project, another labeled dataset is used to maintain privacy. We might remove some of the attributes in the future, if they are not so important for performing the semantic analysis on the dataset. The data obtained using Twitterscraper needs pre-processing, such that the dataset can be used by different ML models as an input.

# 5. PRE-PROCESSING FOR DATASET

Pre-processing plays an important part in the semantic analysis. In order to create a dataset which is fit for using as an input for the given ML models.We use pre-processing techniques to clean the duplicate records and make the dataset fit for the models. We obtained nearly 16000 tweets and we will be working with a sample dataset taken from these 16000 instances.

First step is we need to remove duplicates. This can be done with the hep of user id and tweet id. If we make sure that no two tweets have same user id and tweet id then there can be no repetition of the tweets in the dataset. We can also remove the tweets which is to short to provide any information to the model. For example if a tweet is consists of only emoticons it will be difficult to evaluate the tweet. It might be possible that the emoticons can be sarcastic, in this case it will be difficult to categories the tweet, hence it is optional to remove it.

Second step is annotation of the dataset. We will be performing classification on the Twitter dataset, and as we know in classification techniques we must know the underlying classes. Hence, we must categories the tweets into different classes. Classes in which we will be dividing tweets are Positive or Negative. This will be performed manually, that is, here we will be adding another column at the end of the dataset named Target column and the value of the attribute could be any of the classes discussed above. To remove bias from the annotation task the data was annotated by me and two of my friend and then the final annotation was performed by the majority class after comparing results of all the three annotations. If a tweet was annotated as positive by first annotator, negative by second annotator and positive by third annotator it was finally considered as positive. If a tweet was annotated as positive by first annotator, negative by second annotator and neutral by third annotator or any combination like this it was finally considered as neutral. We remove these neutral tweets from the dataset. Which removes the question of the dataset being bias.

After adding the target column to the dataset. Next step is to process the text of the tweet. In order to predict the underlying class of the tweet we must process the text written in the tweet. So that the model can create correlation between the text of the tweet and underlying class or the target column. This can be done using various methods. The first method we will be using to pre-process the text is known as removing stop words. Stop words are basically simple words in English language they can be 'a', 'an', 'is', 'to' and etc. These words does not provide any useful information of a text being positive or negative. Therefore, these words are of no use to a model for predicting an underlying group. Thus, these words can be removed from the text to increase efficiency of the input which will eventually help to exact efficient output from the model. Another technique used in text mining which is used in semantic analysis to prepare and remove unnecessary data from dataset is stemming. Stemming as the word suggest, it is used convert a given word to its root word. This helps removing acronyms and complex abbreviation that in itself does not have any meaning to the machine but if converted to their root word make a lot of sense and helps the model to identify the target column on the basis of the root word. For example gr8 can be replaced as great. Stemming performs it's operation without considering the context of the word in the sentence. Stemming treats every words as a different individual without considering the context in which the word is used in. Another methods which is similar to stemming can be used to which is known as lemmatization. lemmatization is used to group together different words or group an action or a verb in a single word so that we can identify the action performed just using a single word. For example singing can be replaced by sing. Often Lemmatization and Stemming are considered very similar and they are very similar techniques indeed, but there is a small difference between them , that is, in lemmatization the context of the word is considered while converting it while in stemming a word is converted without considering it's context. In this process we will ignoring the emoticons and punctuation marks. Looking at the length of the project we can use them but it will take a lot of time to convert the meaning of everything written in a tweet and then performing stemming or lemmetization. Hence, we will be ignoring the emoticons and punctuation marks for this project.

| Words | Replaced by |
|---|---|
| Gr8 | Great |
| ttyl | talk later |
| lol | laugh |
| rofl | laugh |

This table shows how can we stemming and lemmatization can help to convert the words which can be easily processed by the models.

To perform stemming or lemmatization we used Stanford Nltk library to convert the words to their respective root words. Another good alternative of the Stanford Nltk could be penn treebank. This library consist words and their root words which can be install in the local directory using python, and which makes this process relatively easy.

# 6. CLASSIFICATION TECHNIQUES

There will be different classification techniques which we will be using for this experiment in order to calculate the accuracy and precision of the model. In classification models we using models we predict the target column and then we compare results of the prediction to the actual value of the target column. Using this we calculate accuracy and the precision of the model. We will be discussing different models that we will be using to carry out out experiment.

## 6.1 Zero Rule Classifier

Zero rule classifier is used as a baseline model to create a benchmark for every other models. While performing comparative analysis you often need a baseline model to compare and contrast every other model to it. This model helps and provide a minimum threshold for every other model. If a model score below the baseline model than it is not a good model for the given dataset.

The Zero Rule classifier is a simple model where in the training dataset, the model will identify the majority class.

Say for example majority class is positive. Then for every testing instance the model will predict the class as positive. Since, we have three target classes the accuracy of the model will be more than 33.3%.

The name of this classifier is zero rule as we apply ground zero rule that is the probability of a records to be a majority class is higher than the probability of a record to be any other class.

## 6.2 Bag of Words Classifier

After the baseline classifier, we will move on to the next classifier which is bag of word(Bog word) classifier. For this classifier we need the vocabulary of the words and the measure or weight of the words.

In this model, first we analyse the text of the tweets. Then we read the processed words and convert them into a word vector. That vector is call Tf-idf vector. This vector uses the words with the higher frequency in the documents and using this vector we predict the class of the testing instance. Fixing misspellings and converting words to there stem is an important process for this model. If not performed these processing techniques it might affect the efficiency and the output of the model.

After identifying the high scoring words we count there frequency. In the TF-IDF vector TF is term frequency which is to calculate the score of the word in the current document. Here document means the tweet. IDF helps us to know the rarity of the word in the given tweet.

To perform the bog word model we uses the python library which helps us to directly convert the tweet into a vector. This helps to ease the process as in tf-idf vector we need to calculate the exponential.

## 6.3 Naive Bayes Classifier

Another classifier, which we will be using to carry out the experiment is Naive Bayes Classifier. This is a probabilistic classification method which is based on Bayes probability. This model predicts the target column by using the Bayes probability. This model is useful as it can be used to predict multi-class classification, also it is considered that every class is conditionally independent. That helps to calculate the conditional probability of the testing instance easily. This model cannot simply take words as input. Since, it is a probabilistic models it works well with numbers and integers. That is the reason why we have to provide a vector as an input for this model rather than the tweet itself. The best vector to use in this model is the Tf-idf vector.

# 7. RESULT

Data pre processing is very important in data mining. If the data is not pre processed there could be a lot of noise. To compare difference between clean and noisy data, word cloud was created to contrast. Fig 4 shows noisy word cloud and Fig 5 shows clean word cloud.

Three different classifier were trained on the clean data set and the accuracy table is displayed below

| Classifier | Accuracy |
|---|---|
| Zero classifier | 50% |
| BOG classifier | 73% |
| Naive Bayes | 58% |

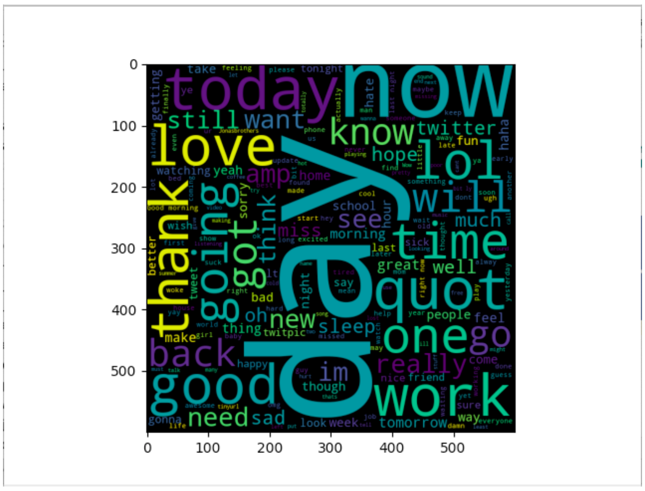

**Figure 4: Word cloud before cleaning**

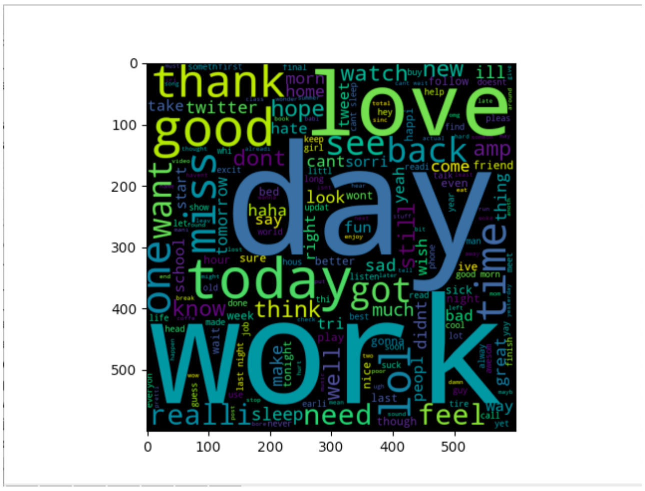

**Figure 5: Word Cloud after cleaning**

In general, a good classifier gives the result of the sentimental analysis in between 70% to 75%. Thus the BOG classifier can be described as a good classifier. For the data set used by Tweepy the best result was also obtained using BOG classifier.

# 8. CONCLUSION

It is very difficult to get a higher accuracy in sentimental analysis, as it totally depends on the data scraped. The data scraped from Twitter and obtained from the labeled dataset consist of many tweets which was difficult to annotate. For example, a tweet ”” can be positive or negative it totally depends upon the perspective of the annotator. It might be neither positive or negative. Many tweets were questions, like ”Anyone up for a movie”, we cannot differentiate this tweet into positive or negative. It is very wise to choose a good dataset for this process.

From classification point of view Bog classifier used Random forest this explains that ensemble classifier can help us get better result in sentimental analysis. This is because, it is creating 200 trees simultaneously and the feature by the best tree is used for the main model.

## 9. FUTURE WORK

The work that will help with the better results in this project is using few more classification techniques like ensemble classifier more. Ensemble classifier as the word suggests is ensemble of the different classifiers. It consists of different classification models. Every classifier learns the data themselves and different from others. Then each of the classifiers vote for an underlying group and according to the output provided by the different models within the classifier, the majority is considered as the prediction for the given testing records.

Another future work would be performing comparative analysis amongst the different classification techniques. Once we carry out experimentation on all the above discussed Machine Learning models, we evaluate the different results by calculating the precision and the accuracy of each model. Using the results we will compare which model performs better as compared to others model for the given set of data.

Future work for this project will be utilizing the information provided by emoticons and punctuation. We will not be analyzing these aspects of the tweets in this project because of the time issue, but this could be potentially used in future. This might change the result for the model, for the good.

## 10. REFERENCES

[1] M. Kanakaraj and R. M. R. Guddeti. Nlp based sentiment analysis on twitter data using ensemble classifiers. In *2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, pages 1–5. IEEE, 2015.

[2] M. Neethu and R. Rajasree. Sentiment analysis in twitter using machine learning techniques. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–5. IEEE, 2013.

[3] A. M. Ramadhani and H. S. Goo. Twitter sentiment analysis using deep learning methods. In *2017 7th International Annual Engineering Seminar (InAES)*, pages 1–4. IEEE, 2017.