**A Review System Using Sentiment Analysis From Tweets**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Mahfuzur RahmanChowdhury  Computer Science and  Engineering  BRAC University  Dhaka, Bangladesh mahfuzur.rahman.c@gmail.com | Wasim Mahmud Surjo  Computer Science and  Engineering  BRAC University  Dhaka, Bangladesh wasimsurjo@gmail.com | Shoaib Ahmed Dipu  Computer Science and  Engineering  BRAC University  Dhaka, Bangladesh shoaibahmeddipu@gmail.com |

***Abstract—A review system we users will share a string and system will give a review about the people’s positive or negative sentiment about that string by analyzing similar type of strings in different tweets by Machine Learning algorithms.***

***Keywords—***opinion mining, service recommendation, text analysis, sentimental analysis.

1. INTRODUCTION

Social media has become an extremely popular medium of communication in worldwide network. Billions of users share their day to day experiences on social media. Therefore, there is no argument that these websites are rich sources of data regarding opinion mining and sentimental analysis. So, in this case machine learning is used to retrieve messages and classify them according to their polarity by showing the results to the users through an interactive interface. It also supports emotional change detection, friend's emotion finding, user's classification according to their massages and statistics among others. As a result, we can build a review system to take data from one of the most popular social media website 'Twitter' and perform Sentimental Analysis.

1. LITERATURE REVIEW o

Humans are always not only obsessed but also influenced by social feedback on almost every aspects of life. Thus, any study or process which acts as a medium of extracting feedbacks are dealt with utter importance. Such is sentimental analysis, where the sentiment of people is analyzed from their words. Sentimental analysis is not a new field of study. Research and improvements in this sector has been going on for a long time. One of the most popular website ‘twitter’ has been a target for analysis in several studies. There is no doubt that twitter is a source of day to day, event to event updates of millions of people all around the world regarding popular events, brands, shows, etc.

There are several algorithms that have been implemented time to time for text and sentimental analysis. SVM (support vector machine), MLP (multilayer perceptron), decision trees, etc. Each algorithm has its own flaws and advantages. SVM proved out to be a very efficient algorithm in several studies. However, SVM is not the only means of good analysis. In this paper, we will us python libraries to carry out sentimental analysis which are based on machine learning algorithms like Naive Bayes and decision trees. These libraries have proven to be extremely well performing over the years.

1. METHODS AND ALGORITHMS

We used Twitter API to collect data. In our project we consider only the tweets as a data which has the same word or words that the user wants to know the review. Our code will calculate the polarity of each tweet of our data and make an average of those. After that we will predict the sentiment of people about that word or test according to our result. A flow chart of our work pipeline is given below. (Fig.I)

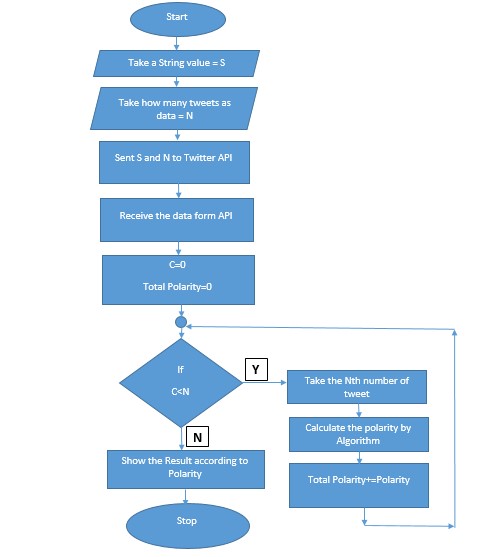


Fig.I: Flow Chart of Work Pipeline

1. *Algorithms*

There are mainly two types of Sentiment Analysis. One is rule based and another is automatic. We our project we used automatic type. Here we used a Machine Learning algorithm named Naiyve Bayes.

1. *Pseudo Code*

* Step 1: Take the String and number of tweet to analysis.
* Step 2: Take data from Twitter API.
* Step 3: Take a tweet and divided the tweets into individual words.
* Step 4: Check frequency of each word.
* Step 5: Check the word is positive or negative from textblob.
* Step 6: Calculate the probability of each word for positive and negative.
* Step 7: Multiply the probabilities of all positive and all negative separately.
* Step 8: Divide those results into by probability of positive and probability of negative.
* Step 9: The probability which is greater is the result
* Step 10: Save the result
* Step 11: If there is tweet remaining then go to

Step 3

* Step 12: Make an average
* Step 13: Show the result

*C. Libraries*

Here, to implement our algorithm we used some libraries of python.

1. We used tweepy library to collect data from

Twitter API.

1. We used re library as a regular expression method library by which we can find the string fast.
2. We used textblob library to get the positive and negative value for those tweets and words. It is a NPL library for python3.
3. We used time library to get rid of “ Twitter error response: status code = 429” error.

IV. RESULT

We predict the data in 7 categories. The relation between the polarity and our category is given below.

1. Strongly Negative: -1<=Polarity<-0.6
2. Negative: -0.6<=Polarity<-0.3
3. Weakly Negative: -0.3<=Polarity<0
4. Neutral: Polarity=0
5. Weakly Positive: 0<Polarity<=0.3
6. Positive: 0.3<Polarity<=0.6
7. Strongly Positive: 0.6<Polarity<=1

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

*A. Test Data*

For determine the f1 score, precision, recall, accuracy, ROC curve we made a test data file which content some name of movies, animes, player names and also labeled with the true value weather it is good or bed.

After that we predict the value for each data and check our prediction value with the true value. In this way we calculate the f1 score, precision, recall, accuracy, ROC curve. The results are given below.

Table.1: Confusion Matrix

|  |  |  |
| --- | --- | --- |
|  | Positive (P) | Negative (N) |
| True (T) | 69.8630 | 2.7397 |
| False (F) | 19.1780 | 8.2197 |

Accuracy = (TP + TN) / (P + N)

= (69.8630+2.7397)/(100)

= 0.7808

[Precision=](http://onlineconfusionmatrix.com/#measures)TP / (TP + FP)

= 69.8630/(69.8630+19.1780)

= 0.7846

Recall=TP / (TP+FN)

= 69.8630/(69.8630+8.2197)

=0.8947

F1 Score= 2\*((Precision \* Recall) / (Precision +

Recall)) =2\*((0.7846\*0.8947)/(0.7846+0.8947)) = 0.8360.

ROC curve in Fig.II :

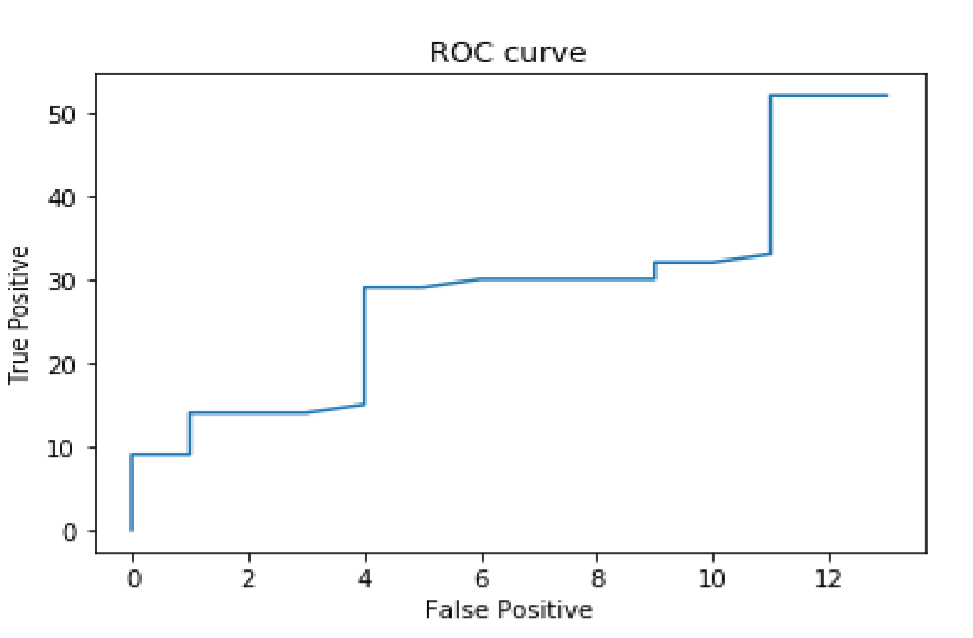


Fig.II: ROC Curve

*B. Comparison*

Our work is different than others because we focus on recall value. We did not focus on accuracy or precision. In our review system there is very low chance that we are telling a negative thing that it is positive. Therefore, people who will use our system will not have to face a negative thing because of our positive review. Our system is also better because we are using recent data direct from twitter but most of the system uses a fixed dataset.

Therefore their data will be out dated soon.

.V. Discussion

Our recall value is higher than precision. It indicates the class is correctly recognized. Our model has values of accuracy & precision which is less than recall value. Using our own labeled dataset, we could not reach to 80% of accuracy because we made the true label according to some website and some of our own review. Those data were not always not. Alongside with this, we were having problems to find more negative words for our data set. Because of that reason 65% of the our data pf dataset were positive and 35% were negative.

1. Conclusion

The biggest problem we face to do our project is “ Twitter error response: status code = 429” error. This is an error when we give too many requests to a server in a very short amount of time. As we are using Twitter API as our server, when we were testing our data it gives this error after every 25 search. To solve this problem we import time library to give a break after every 20 search in the twitter API.

As a limitation, to calculate the f1 score, precision, recall, accuracy, ROC curve our code will give the result very slowly because of the time.sleep() method. And it will also be difficult for us to test our code on a large dataset.

Our future plan is to build a user interface to use our code more easily.

1. REFERENCES.

1. [https://github.com/thejavapocalypse/Twitter-Sentiment-Analysis](https://github.com/the-javapocalypse/Twitter-Sentiment-Analysis)
2. *Cai, K., Spangler, S., Chen, Y., & Zhang, L. (2008). Leveraging Sentiment Analysis for Topic Detection. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent*

*Agent Technology, 1, 265-271.*

*10.1109/WIIAT.2008.188*

1. *Gamon, M., Aue, A., Corston-Oliver, S., & Ringger, E. (2005). Pulse: Mining customer opinions from free text. Lecture Notes in Computer*

*Science, 36(6), 121–1324.*

*doi:10.1007/11552253\_12*

1. *Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In Proceedings of the Tenth ACM SIGKDD International conference on Knowledge discovery and Data mining (pp. 168-177), ACM.*
2. *Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in*

*Information Retrieval, 1(2), 1–135.*

*doi:10.1561/1500000011*

1. *Prabowo, R., & Thelwall, M. (2009). Sentiment Analysis: A Combined Approach. Journal of Informetrics, 3(2), 143–157. doi:10.1016/j.joi.2009.01.003*
2. *Rajeev, P. V., & Rekha, V. S. (2016). Opinion Mining of User Reviews Using Machine Learning Techniques and Ranking of Products Based on Features. Advances in Intelligent Systems and Computing., 39(8), 78–85.*
3. *Sharma, R., Nigam, S., & Jain, R. (2013). Supervised Opinion Mining Techniques:A Survey. International Journal of Information and Computation Technology, 3(8), 737–742.*