Polarity Testing and Analysis of tweets in Twitter using Tweepy

Rahul Pandya[1], Sujal Charak[2], Suraj Moolya[3], Ritish Dahivalkar[4] Hardik Gadhadara[5]

[1,3]Department of Electronics and Telecommunications,
MCT's Rajiv Gandhi Institute of Technology, Mumbai, Maharashtra, India,
[2,4,5]Department of Electronics and Telecommunications,
Atharva College of Engineering, Mumbai, Maharashtra, India.

ABSTRACT
Sentiment Analysis has been a very important part of analytics for data scientists over the years. It has been a very detailed and an important area of research and development which enables the user to find the acknowledgement factor for the area of interest. Social media is always evolving and the most interactive media of individual communication and broadcasting. Sentimental analysis of is the best alternative for peer reviewing in terms of a certain criterion. This paper deals with an analytic study over a twitter based dataset which involves pulling of certain number of tweets using API linking and then performing the polarity check on the number of tweets pulled with respect to that particular keyword. An approach involving unsupervised machine learning algorithms along with natural language processing generates significant results in the task over the traditional lexicon method used.

KEYWORDS: Machine Learning, Polarity Testing, Python, Sentiment Analysis, Twitter

Date of Submission: 28-03-2021 Date of acceptance: 11-04-2021

I. INTRODUCTION

Sentimental Analysis has been a very in depth statistical studying topic in computer science and in developing fields. It has been a scope of study in data mining especially. In the earlier times, people relied upon the opinions of only the closer neighbours, that is, friends, family and relatives which used to form only a smaller group of people. The campaigners made it t the larger part of audience through polls. The continuously evolving social media has now been the largest platform for opinion generation on topics in terms of relative review on personalised as well as mass areas of interest. This is a very developing opportunity as it involves organizing, classifying and detection of the relativeness between the particular opinions. Sentiment analysis is a very important aspect for organizations and big firms so as to achieve the review results for performance of their specific trademarks in terms of business queries.

Performing sentimental analysis is a challenging task. It is because of the pattern and nature of the text that has to be imported. Social media gives the researchers a very large data to access and perform analytics on as it is the biggest platform of interaction. So the amount of text and opinion extraction on such platforms is a demanding task. The linguistic barrier is one of them. However, by using natural language processing and machine learning algorithms these obstacles can be overcome. The opinion and emotion tracker is the most significant of all sections which involves the categorical word based classification of the texts and then assignment of the polarity to them accordingly. The accuracy of prediction can then be calculated and improved significantly based on the model parameters. In this paper, the authors have performed a statistical polarity check of extracting a particular number of tweets from the social media platform Twitter. A definite keyword is used as a parameter to pull tweets related to that and then a check and measure is based on natural language processing along with classification is applied to generate a plot of distribution of favour and negative tweets.

The tweets can be categorized into three major sectors based on the criterion. Positive, strongly positive and weakly positive for tweets in support of the keyword. Negative, weakly negative and strongly negative for tweets not in favour of it and neutral for no such bias intended. The performed sentiment analysis can be performed on many platforms. For example, Movie suggestions or reviewer in Netflix and Hotstar, Restaurant suggestion and reviewer in applications like Zomato and Swiggy, Similar music suggestion based on genre and recent plays in Spotify, Gaana etc.

II. SYSTEM BLUEPRINT

Sentiment analysis requires almost negligible costing. Our system involves using of a scripting language like Python or R along with the installation of an easy to use IDE. Here we have used Jupyter Notebook which comes built in with Anaconda Software [1]. Visual Studio and Pycharm could also be used. System processors having i3 intel processors and above versions with a basic graphic card is sufficient to run the libraries and ML algorithms. We suggest using Anaconda as it enables the user to download the whole bunch of libraries and software as a bundle and other required ones can be easily downloaded using a simple and efficient command known as pip.

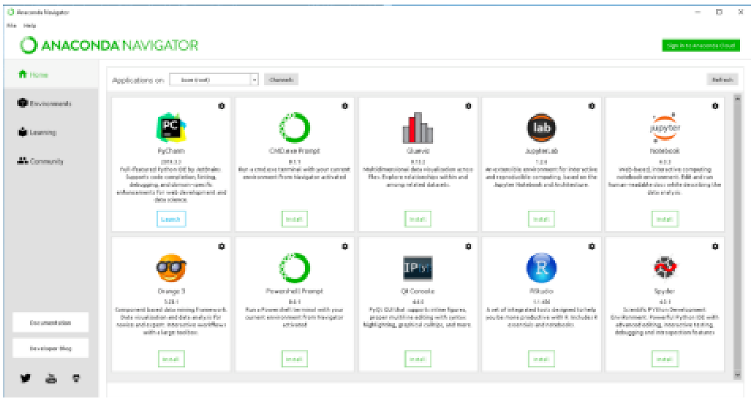


Figure: 2.1 Anaconda Software

The following are the basic requirements to perform sentiment analysis:

- Tweepy, a basic python library for accessing the twitter API [2].
- Pandas – it is used to generate the CSV for the pulled and recorded tweets.
- Numpy package – it is used for numerical calculations in python library.
- Seaborn – it is a library used for statistical data visualisation [3].
- NLTK word tokenizer **-** A tokenizer that divides a string into substrings by splitting on the specified string (defined in subclasses) [4].

III. METHODOLOGY ADOPTED

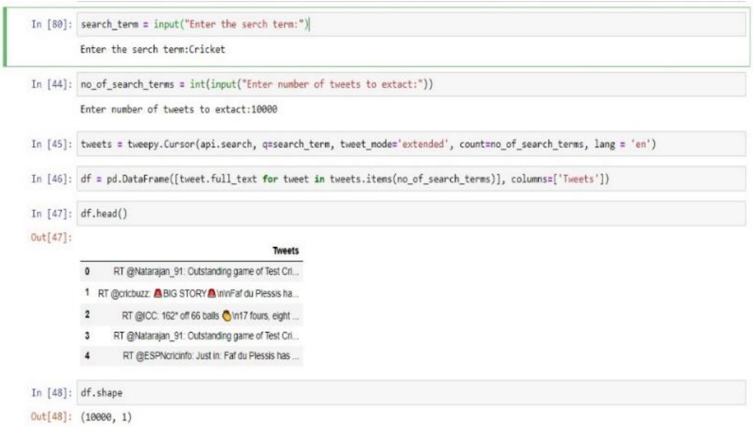
There is a lot of information that needs to be imported from the social media platform. Twitter generates billions of tweets per year across approximately 187 million users. These tweets contain a diverse information on all sorts of topics across the globe and are multi lingual. Pulling data across such an active platform is a very exciting stuff to do. One can access almost all kinds of topics and generative study of them and get to know about the current affairs and get updated whilst putting his or her opinion on it. Regardless of these, sentiment analysis of social media platform like twitter gives a public opinion easily. Just mentioning that topic can generate thousands of results related to that topic. Here, we are going to perform a similar experiment. We shall be pulling a specific number of tweets related to a topic. The topic here shall be the keyword defined parameter.

3.1 API Authentication and Keyword Defining.

Firstly, we need to import all the libraries required to perform the task. We import the libraries of numpy and pandas to generate the csv for maintaining the record of the pulled tweets and keep a count. We import 're' that is regular expression which specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing) [5]. After the libraries are imported we create a csv file to save the tweets that we are about to pull.

Next is the API authentication via Tweepy. We need to connect or give access of our twitter account to the program we are building. This needs to be done through API authentication using an interface application. Twitter has an inbuilt developer application interface for API generation which can be used programmatically for analysing and retrieving data [6]. Once the authentication is done one can then start accessing twitter media via program. We take the user input to the program as a keyword to be searched on. The keyword is the topic or the area of interest on which the user intends to perform a research on. Also we need to specify the amount of tweets in text that we need to pull with respect to that keyword.

Here, in Figure 3.1 we can see that we defined the keyword as ‘Cricket’ and the total number of tweets to be pulled related to that keyword as thousand. The tweets pulled shall only be in English as the language parameter defined here is English. This can be changed as per user requirement.

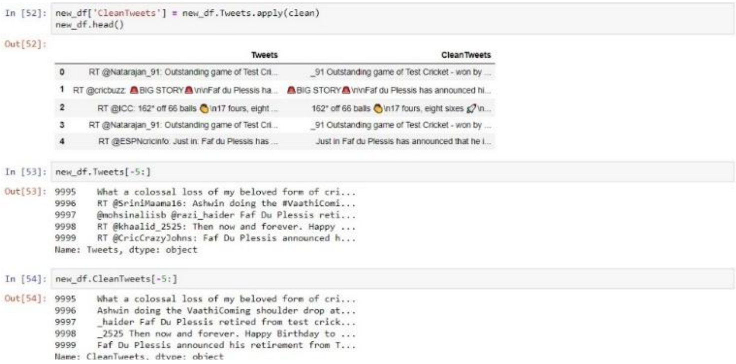


```
In [80]: search_term = input("Enter the serch term:")

         Enter the serch term:Cricket

In [44]: no_of_search_terms = int(input("Enter number of tweets to extact:"))

         Enter number of tweets to extact:10000

In [45]: tweets = tweepy.Cursor(api.search, q=search_term, tweet_mode='extended', count=no_of_search_terms, lang = 'en')

In [46]: df = pd.DataFrame([tweet.full_text for tweet in tweets.items(no_of_search_terms)], columns=['Tweets'])

In [47]: df.head()

Out[47]:
                                          Tweets
         0     RT @Natarajan_91: Outstanding game of Test Cri...
         1  RT @cricbuzz: 🚨BIG STORY🚨\n\nFaf du Plessis ha...
         2        RT @ICC: 162* off 66 balls 🏏\n17 fours, eight ...
         3     RT @Natarajan_91: Outstanding game of Test Cri...
         4        RT @ESPNcricinfo: Just in: Faf du Plessis has ...

In [48]: df.shape

Out[48]: (10000, 1)
```

Figure: 3.1 Tweets Stored with respect to keyword.

3.2 Importing Tweets and Cleaning

After the tweets are pulled next step is cleaning of the text. Here, by cleaning we mean to remove the unnecessary text terms such as hashtags, hyperlinks and other such symbols present in the text of the tweet and convert it into plain text containing only alphabets. In Figure: 3.2 We can observe the tweets before and after cleaning them.



```
In [52]: new_df['CleanTweets'] = new_df.Tweets.apply(clean)
         new_df.head()

Out[52]:
                                          Tweets                                       CleanTweets
         0     RT @Natarajan_91: Outstanding game of Test Cri...    _91 Outstanding game of Test Cricket - won by ...
         1  RT @cricbuzz: 🚨BIG STORY🚨\n\nFaf du Plessis ha...  🚨BIG STORY🚨\n\nFaf du Plessis has announced hi...
         2        RT @ICC: 162* off 66 balls 🏏\n17 fours, eight ...      162* off 66 balls 🏏\n17 fours, eight sixes 🚀\n...
         3     RT @Natarajan_91: Outstanding game of Test Cri...    _91 Outstanding game of Test Cricket - won by ...
         4        RT @ESPNcricinfo: Just in: Faf du Plessis has ...     Just in Faf du Plessis has announced that he i...

In [53]: new_df.Tweets[-5:]

Out[53]: 9995     What a colossal loss of my beloved form of cri...
         9996     RT @SriniMaama16: Ashwin doing the #VaathiComi...
         9997     @mohsinaliisb @razi_haider Faf Du Plessis reti...
         9998     RT @khaalid_2525: Then now and forever. Happy ...
         9999     RT @CricCrazyJohns: Faf Du Plessis announced h...
         Name: Tweets, dtype: object

In [54]: new_df.CleanTweets[-5:]

Out[54]: 9995     What a colossal loss of my beloved form of cri...
         9996     Ashwin doing the VaathiComing shoulder drop at...
         9997     _haider Faf Du Plessis retired from test crick...
         9998     _2525 Then now and forever. Happy Birthday to ...
         9999     Faf Du Plessis announced his retirement from T...
         Name: CleanTweets, dtype: object
```

Figure: 3.2 Cleaned Tweets

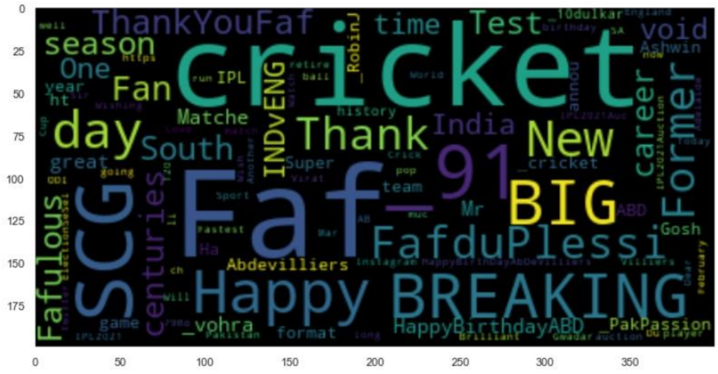


Figure: 3.3 Wordcloud

We can then test the subjectivity and polarity of the cleaned tweets. In this example we received a total polarity sum of 1663.5242. Plotting of wordcloud helps us to visualize that the tweets pulled have been related to the keyword parameter defined. Generating a wordcloud is a simple task involving tokenizing all the words received in the tweets and them plotting a visualizing figure containing randomized number of words.

Figure 3.3 shows a word cloud for a hundred words in randomized tweets from the example used in this paper.

3.3 Testing Polarity and Results

The final stage involves the polarity testing of the tweets. This is the basic plotting of how the people are reacting on the searched term, that is, the defined keyword with respect to the number of searched terms, that is, the tweets. Polarities can be classified as mentioned earlier into three parts of negative and positive. Tweets with polarities in the range of 0 to 0.30 lie in weakly positive/weakly negative, tweets within the range of 0.31to 0.60 in the positive/negative and tweets in the range greater than 0.61to 1 are considered as strongly positive or strongly negative respectively. The polarity gets defined on the basis of the negation parameters like not, nor and such disregarding words present in the tweet's sentences. This is done natural language processing. One such experiment of finding the negative speech recognition within the tweets in the paper Deep Learning for Hate Speech Detection in Tweets [7].

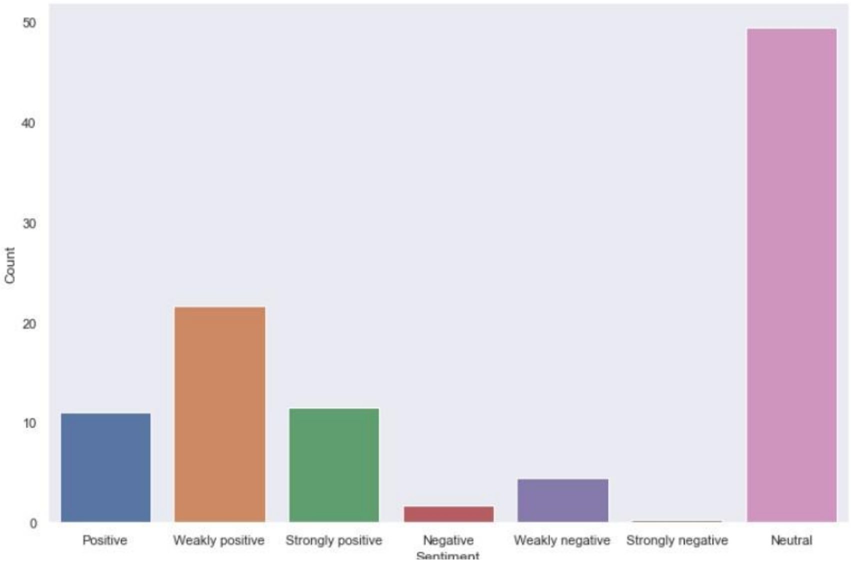


Figure: 3.4 Plot of Sentiment Analysis

The Figure 3.4 shows the polarity plot for the experiment performed. It can be observed from the bar plot that the neutral tweets with respect to the current Cricket norms are huge followed positive tweets.

The following was the count percentage of the sentiment analysis:

- Neutral: 49.46
- Strongly Positive: 11.52
- Positive: 11.07
- Weakly Positive: 21.57
- Strongly Negative: 0.22
- Negative: 1.66
- Strongly Negative: 4.43

IV. CONCLUSION

This paper provides a basic evaluation on performing sentimental analysis in an easy approach. We achieved a bar plot for a thousand tweets for the keyword 'Cricket' and got a very positive response. This response varies with respect to the number of tweets imported and also with respect to the instant at which the following analysis is performed. A larger number of tweets sets a higher order loading capacity and takes time to perform the analysis but the polarity generated would be of larger opinion factor with respect tot the masses.

With respect to possible modifications, this method is expandable and flexible as the user can make multiple comparisons of multiple keywords using the similar approach and get a statistical plot for the same. Using of SVMs (Support Vector Machines) makes this a longer but more efficient analysis.

Applications of sentimental analysis is a very vital ingredient in business and for growing companies. They can keep a track on the user or customer indulgence of their products and how they are performing in the competitive market. Using of decision trees and regressions can improve the analysis features up to the prediction levels. Future scope involves using GANs (General Adversarial Networks) for sentiment analysis.

REFERENCES

[1]. Anaconda Software. https://www.anaconda.com

[2]. Tweepy Python Library. Available at: https://docs.tweepy.org/en/latest

[3]. Seaborn Library for Statistical Visuals. https://seaborn.pydata.org

[4]. NLTK word Tokenizer. Available at: https://www.nltk.org/api/nltk.tokenize.html

[5]. Regular Expression. Available at: https://docs.python.org/3/library/re.html

[6]. Twitter Developer API Function. https://developer.twitter.com/en/docs/twitter-api/getting-started/guide

[7]. Deep Learning for Hate Speech Detection in Tweets, Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, Vasydeva Varma. https://arvix.org/abs/1706.00188

Rahul Pandya, et. al. "Polarity Testing and Analysis of tweets in Twitter using Tweepy." *International Journal of Computational Engineering Research (IJCER*), vol. 11, no.2, 2021, pp 06-10.