



Implementation of Sentiment Analysis using Python and Flask

Khushi Agarwal¹, Ishita Kapoor², Dr.Pankaj Kumar³

Btech, Computer Science and Engineering

Shri Ramswaroop Memorial College of Engineering and Management

Lucknow, India

Abstract—With the growing technology and its easy access, social media has become a part of everyone's life wherein they are free to voice their opinions publicly on any given topic. One such platform is Twitter, wherein people comment upon a huge variety of subjects ranging from social activities to any remark on products. With millions of users tweeting every day, it can be challenging to make sense of the vast amount of data generated. There can be positive, negative, or neutral opinions that tend to confuse people until correctly demonstrated. Sentiment analysis is a powerful tool for analyzing this data and extracting insights from it. In this paper, we present a Flask-based web app that uses TextBlob in order to perform sentiment analysis on the tweets to segregate them and give a pictorial representation of the review of the masses on the topic.

Keywords—Twitter, Sentiment Analysis, TextBlob, Flask, Python

I. INTRODUCTION

Social media has grown tremendously as a platform used to freely share opinions on a wide variety of subjects. One such booming social networking platform is 'Twitter' wherein people comment on topics ranging from social activities to product reviews[4]. The online communities provide an interactive forum for consumers to inform and influence others. Spurred by this growth, companies and media organizations are increasingly seeking ways to mine information to get people's perspectives on their products and services[7]. These insights can determine the behavior and decision-making patterns of consumers, allowing organizations to stay updated with their customers' desires and improvise when required.

This can be achieved through Sentiment Analysis which is one of the most emerging fields in Natural Language Processing. Sentiment Analysis is a process in which the given text can be classified as positive, neutral, or negative. Sentiment analysis plays an important part in social media such as Twitter and Amazon. It is used to get the sentiment value of the text and classify it whether it was positive, negative or neutral. Social media is a vast collection of unstructured data[1]. Owing to this fact, there is a need to clean and classify the data in order to make it easy to process. Subsequently, we perform sentiment analysis on the tweets and display the results in a graphical format, classifying the customers' viewpoints via tweets into positive, negative, and neutral.

The rest of the paper is as follows:- Sect2 contains the information retrieved from previous papers and research. Sect3 talks about the proposed algorithm which is being used for the project, Sect4 contains the implementation of the project along with the libraries and methods that are being used for the analysis of the tweets. It also contains the result and its representation. Sect4 represents our findings and scope for future improvement.

II. LITERATURE REVIEW

Sentiment analysis has been studied extensively in the field of natural language processing with research ranging from document-level classification to computation of the polarity of words and phrases. Researchers have developed a wide range of techniques for sentiment analysis, including machine learning algorithms and lexicon-based approaches. It becomes difficult to identify the nature of a tweet because of the restriction of its size but there are various authors who have given their insights regarding sentiment analysis using natural language processing.

A. Kumar et al.[1] in their paper talked about how sarcasm can be a big hindrance in determining the sentiment of the tweet. Moreover, they talked about the two forms of sarcasm which are numerical and contextual. The methods of removing it using the word cloud of sarcasm words have also been demonstrated.

Shanshan Yi, Xiaofang Liu, et al. [2], analyze customer reviews and determine the sentiment of customers towards a product. Created a hybrid recommendation system using a machine learning regression model. The performance of the model was analyzed using three matrices namely MAE (mean absolute error), MSE (mean squared error), and MAPE (mean absolute percentage error). The future scope of extending their work on customer interest across different geographical locations has also been stated.

Nirag T. Bhatt, Asst. Prof. Saket J. Swarn deep et al. [3], in their paper, demonstrate sentiment analysis, levels of sentiment analysis, advantages and disadvantages of sentiment analysis, and its application. Their main focus is on different feature extraction techniques and how they can be performed through different machine-learning approaches.

III. PROPOSED ALGORITHM

We have created a Flask based web app which consists of multiple pages: a login page wherein the users can register or log-in , a home page (as shown in Fig2.1) where users can enter the keyword and number of tweets which they want to be analyzed, and a results page where the sentiment analysis results are displayed. The app uses the Tweepy library to access the Twitter API and retrieve real-time tweets containing the user-specified query. This can be done only after creating a Twitter developer account and obtaining the necessary API keys and access tokens for authentication.

The tweets fetched cannot be used directly for the analysis, thus pre-processing of them is being done through which all the unnecessary data is being removed.

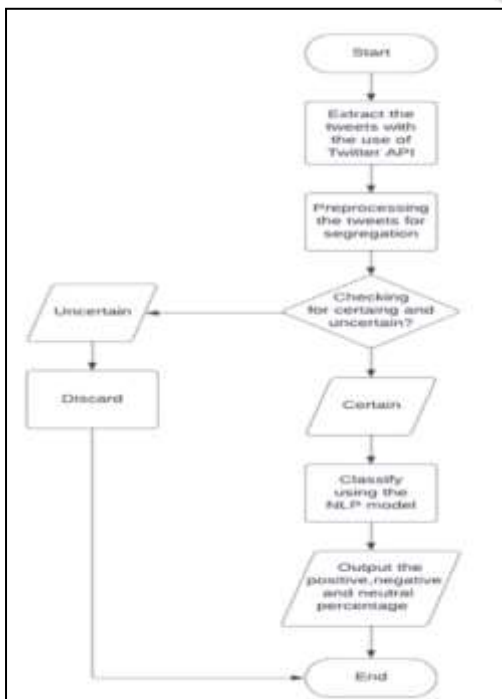
Then the TextBlob library is used to calculate the average polarity score of the tweets fetched. The analyzed sentiment results along with other relevant information is stored in a Pandas dataframe for further analysis. Finally, the app uses the data visualization library Matplotlib to create plots (e.g., bar chart, pie chart, or line chart) to display the distribution of positive, negative, and neutral tweets in a visually appealing way. The generated plots along with other resulting information are rendered in the Flask web app to display them to the users. The web app can be hosted on a server, making it accessible to users from anywhere with internet connectivity.



Fig.2.1 Snapshot of the web app

IV.IMPLEMENTATION

To implement the sentiment analysis on Twitter data, our project makes use of the following modules:-



A.

DATA ACQUISITION

A data set is needed for sentiment analysis. As a user would want to gather insights about a particular topic, we have created a user interface web app wherein the user can log in, thus protecting their privacy, and can then enter any keyword or topic along with the number of tweets of which he/she wants the analysis to be done. After that, the raw tweets which are the most relevant will be fetched from Twitter using the Tweepy API. Since the tweets fetched are unstructured and contain a lot of irrelevant information thus they are sent in for the data preprocessing step.

B. DATA PREPROCESSING

The raw tweets thus extracted are not suitable to process directly because of the causal nature of the language used on social media. Therefore, preprocessing of the data is required in order to make it suitable for the analysis and to improve the efficiency of further steps. The preprocessing involves the following steps:-

1) Removal of URL

The tweets may contain various links that are irrelevant to the sentiment analysis and consequently needs to be removed from the text. The URL is being removed using the python library re(regex).

Example:- The link “https:..” is being removed and the rest of the tweet remains the same.

2) Removal of @mentions

The mention of the username or handles is also irrelevant in the analysis of sentiment. Thus the @user annotation is being removed. Since all the usernames will start from ‘@’ thus python library regex is being used for it.

3) Removal of punctuations

Since punctuations don’t play an important role in sentiment analysis, thus they are being removed using the python library regex to clean the data.

Example- A tweet after preprocessing would look like:-

Gautam Gambhir said LSG is a perfectly balanced team without KL Rahul

Thus, after preprocessing we have the extracted tweets which can be easily used for the analysis.

C. ANALYSIS

After the preprocessed tweets are received, their sentiment needs to be analyzed for which they are being stored in the form of a table using the pandas library available in Python for easy access to each of the tweets. The sentiment of the tweets is then analyzed, using Textblob, depending on their subjectivity and polarity. TextBlob uses a rule-based approach for sentiment analysis, where it assigns polarity scores to words based on a pre-defined lexicon of words and their associated polarity values. The working of TextBlob for sentiment analysis can be broken down into the following steps:

1) Sentiment polarity calculation:

This step is to calculate the sentiment polarity of the text using the ‘sentiment.polarity’ property of the TextBlob object. The sentiment polarity can be either a positive ,negative or zero value.The sentiment polarity is

calculated by taking the average of the polarity scores of each sentence in the text.

The polarity score of a sentence is then calculated as the sum of the polarity scores of its constituent words, normalized by the square root of the number of words in the sentence. This normalization is done to prevent longer sentences from having a higher polarity score simply because they contain more words.

The formula used by TextBlob to calculate the sentiment polarity of a sentence is as follows:

$$sentiment_polarity = \frac{\sum(polarity(word) \text{ for word in sentence})}{\sqrt{(len(sentence))}}$$

2) Subjectivity calculation:

In addition to sentiment polarity, TextBlob also provides a 'sentiment.subjectivity' property that calculates the degree of subjectivity in the text. The subjectivity ranges from 0 to 1, where 0 represents objective text and 1 represents subjective text.

After the values of subjectivity and polarity are received, the analysis is done as follows:-

- If the subjectivity/polarity ratio is >0 then the tweets will be regarded as positive,
- If the subjectivity/polarity ratio is <0 then the tweets will be regarded as negative,
- If the subjectivity/polarity ratio is =0 then the tweets will be regarded as neutral.

The classification of each tweet is then stored in the dataframe so that they could easily be represented in graphical format.

D.RESULT AND VISUALIZATION

After the sentiment of the tweet is analyzed, it is essential to create a pictorial representation for easy comprehension by the user. This allows a clear and intuitive representation of the sentiment distribution. Hence, here the data analyzed is being displayed in the form of two charts: a bar chart and a pie chart. The charts displayed on the results page showed the percentage distribution of positive, negative, and neutral tweets for each query.

We have labeled the tweets in three classes according to sentiments observed in the tweets: positive, negative, and neutral. From the results received, the following can be deduced about the labeling process:

Positive: If the entire tweet has a positive/happy/excited/joyful attitude or something is mentioned with positive connotations. Also if more than one sentiment is expressed in the tweet but the positive sentiment is more dominant.

Negative: If the entire tweet has a negative/sad/displeased attitude or something is mentioned with negative connotations. Also if more than one sentiment is expressed in the tweet but the negative sentiment is more dominant.

Neutral: If the creator of the tweet expresses no personal sentiment/opinion in the tweet and merely transmits any information that would help to analyze the sentiment then it will be regarded as a neutral tweet. Advertisements of different products would be labeled under this category.

The results shown in the figures were fetched when the keyword entered by the user was 'competitive

programming' and the number of tweets were 100 . Fig 4.2 illustrates a pie chart showing that 25% of the tweets were neutral, 70% were positive, and 5% were negative and the same is being displayed by bar chart(Fig.4.3).

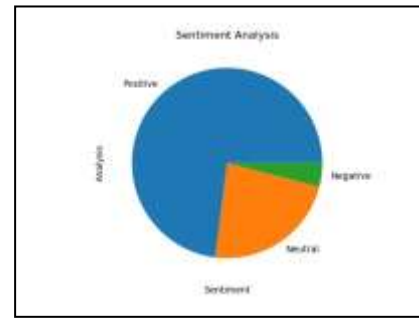


Fig.4.2 Pie chart

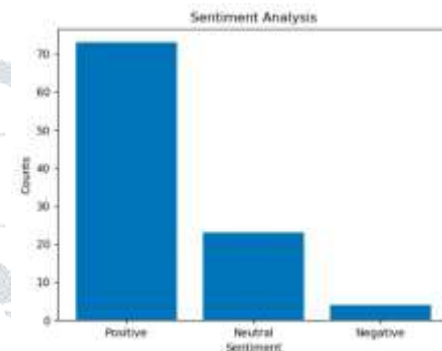


Fig. 4.3 Bar chart

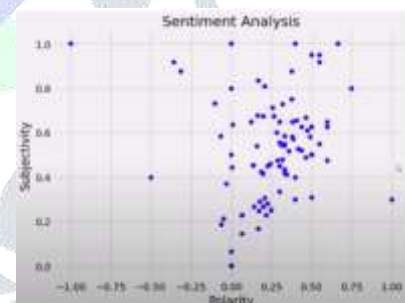


Fig.4.4 Polarity-Subjectivity Chart

A graph of subjectivity and polarity is also displayed for better understanding which is shown in Fig.4.4.

We tested the application by entering several queries related to current events and popular topics. The app was able to fetch tweets containing the queries and perform sentiment analysis on them. The pie chart displayed on the results page showed the distribution of positive, negative, and neutral tweets for each query. Additionally, the average polarity score was shown, providing a general understanding of the sentiment represented in the tweets.

IV.CONCLUSION and FUTURE SCOPE

In this research project, we proposed a web app using Python and Flask for performing sentiment analysis on Twitter data. It provides an easy-to-use interface for users to fetch tweets related to a topic of interest, preprocess the tweets, and analyze the sentiment. If the tweet shows a positive/happy attitude then it is labeled with positive connotations, if it shows a negative/depressed attitude then

it is labeled with negative connotations else the tweet is neutral. The sentiment analysis results are visualized in the form of graphs making it easier to grasp the sentiments of the users.

The app can be used by businesses, researchers, and individuals to gain insights into public opinion on a particular topic. For example, The software might be used by a company to track customer opinions about its goods or services, or by a researcher to gauge public mood towards a particular social issue. It is also convenient and practical, making it simple to use for anyone, including those without technical knowledge. Sentiment analysis of Twitter data is an invaluable tool for understanding public opinions and sentiments about different matters. The project seeks to advance the field of sentiment analysis by offering a workable implementation of a web app that can be used for sentiment analysis on any public opinion.

In terms of future work, the project can be extended to include the following:

- Our application currently works only for Twitter data, and in the future we can expand the same to work upon different data which is being fetched from platforms like Facebook, Instagram.
- TextBlob is a simple and straightforward library for sentiment analysis, but it might not always provide accurate results. The accuracy of sentiment analysis can be improved by using further improved techniques like Deep Learning.
- The performance of the analysis can further be improved by using a real time streaming data processing pipeline which can handle a continuous stream of tweets, allowing for more accurate results.
- A recommendation engine that makes suggestions for what to do based on the sentiment analysis results can be added to the application. Based on the results, for instance, a business might get suggestions which would help him in increasing customer satisfaction.

Overall, our project provides a simple yet efficient method for sentiment analysis on Twitter data.

REFERENCES

1. Atul Kumar, Pooja Agrawal, Ratnesh Kumar, Sahil Verma, and Divya Shukla: "Sarcasm Detection using SVM" *Cognitive Informatics and Soft Computing*(2022).
2. Qin, Zhou, Fang Cao, Yu Yang, Shuai Wang, Yunhui Liu, Chang Tan, and Desheng Zhang. "CellPred: A behavior-aware scheme for cellular data usage prediction." *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, no. 1 (2020): 1-24.
3. Patel, Alpna, and Arvind Kumar Tiwari. "Sentiment analysis by using recurrent neural network." In *Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE)*. 2019
4. . Ahmad, "how much data is generated every minute info," *Social Media Today*, Jun 2018. O. Alonso, K. Shiells, H. J. Lee, and C. Carson, "System and method for generating social summaries," Jan 2019
5. Bhumika Gupta, Monika Negi, Kanika Vishwakarma, Goldi Rawat, Priyanka Badhani: *Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python*,2017
6. Aliza Sarlan, Chayanit Nadam, Shuib Basri: *Twitter Sentiment Analysis*,2014
7. A. K. Jose, N. Bhatia, and S. Krishna, "Twitter Sentiment Analysis". *National Institute of Technology Calicut*,2010.