Sentiment Analysis for Social Media Using NLP

Amit Tiwari [1], Yash Gaidhani [1], Gayatri Katare [1], Khushi Mehta [1], Dr. Mukesh M Raghuwanshi [2]

[1] UG Students, [2] Assistant Professors, Computer Engineering Department
G. H. Raisoni College of Engineering & Management, Pune

ABSTRACT

In today's world Social Networking Platforms play a major role in influencing people and shaping their mindset. They bring out large virtual space for all to express, share their opinions and to influence others through them. In this project we are trying to provide a simple solution for the social media users to get the information and the Sentiment Polarity about particular topic or trend, derived from the tweets itself by the means of Android and IOS Application with an objective of minimizing the spread of fake content or misinformation. Sentiment Analysis and Text Summarization are the two major techniques used in this project with the help of various NLP tool kits.

Index Terms- Text Summarization, NLP, Sentiment Analysis, Social Media, Machine Learning.

1. INTRODUCTION

The analysis of the user sentiments and their summarization can provide a healthy solution for analyzing the varied opinions of different users on a topic. With Increasing use of social media , especially Twitter by users to state their opinions on various Educational, Economical, political topics it provides easy and simple option for developers and analysts for data collection as per the requirement for further study. This project is aimed at studying the sentiment for the topic "Exams during Pandemic" so as to get the overall sentiment of the users at the same and summarize their opinions to get in detail information about the topic and provide the same to all users through a simple Mobile User Interface.

2. APPROACH

To derive the in detailed summery and the sentiment polarity for the selected topic, initially we did develop a simple Mobile Application which would act as a medium for users to access the Sentiment analysis and the Text Summarization Result of the tweets collected with English as a primary Language through search API. The collected data was then preprocessed and analyzed to obtain the required summarized output.

`

3. Data Collection

We choose English as the primary language for selection of the tweets for simplification and thus accessing greater user base for this project. "Exams during Pandemic" was the trending topic among Students, college Officials and professionals and Government Educational bodies discussing about the varies solutions for the University Exams in the situation of Covid 19 making it difficult for students to travel to their respective Colleges and University for the same. The Savitribai Phule Pune University (SPPU) as being one of the oldest University in India was the center of the discussion thus covering majority of Colleges in Maharashtra.

There was mixed opinions from various sectors about the mode of examination and the decision of the Exam cancellation. Online mode of examination was considered by many as a solution, whereas many people consider that Cancellation was the only way considering the technical problems at some points and places and the half Completed academic Syllabus. To analyze this topic the data was collected from the Search API in an unbiased way to provide more realistic output for the same by using some of the top trending topics and tags at that time for the subject.

4. USER INTERFACE

The Main Idea was the give user access to the analyzed tweets and get the total information based on people's opinions for the topic. So we decided to provide a mobile application for the it which would be providing the analysis results in proper way to the users. For this purpose React Native was selected was major Technology stack for Android and IOS application development. The application interface was designed in a way to provide simplicity to the users and easy accessibility.

Initially it would list down the selected topics and tags related to the topic and then would provide the detailed analysis result including the sentiment analysis and text summarization along with some graphical representation about the polarity of the sentiment making it easy to judge the overall polarity of the topic.

The analysis results would be accessed through an API request to the server, whose result would be displayed in proper readable format.

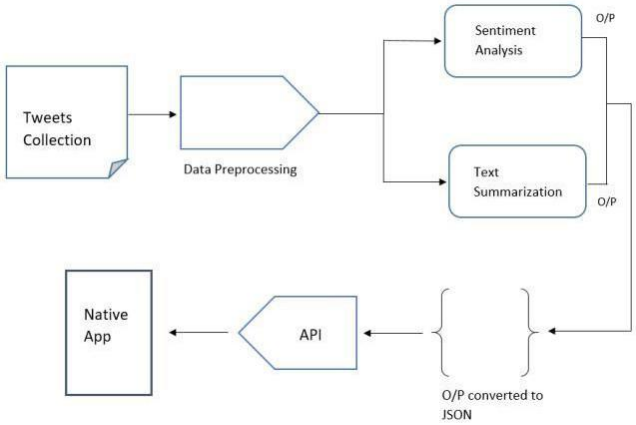

Fig 1 – Project Workflow diagram

5. Sentiment analysis

Sentiment analysis is basically the process of detecting positive or negative sentiment in text. It's often used by businesses to detect sentiment in social data, measure brand reputation, and understand customers. Since customers and users express their thoughts and feelings more freely than ever before, sentiment analysis is becoming an essential tool to monitor and understand that sentiment. Automatically analyzing customer feedback, such as opinions in survey responses and social media conversations, allows brands to learn what makes customers happy or frustrated, so that they can update products and services to meet their customers' needs.

5.1 Technologies and Tools Used for Sentiment Analysis

Sentiment Analysis came about after **Text Mining**. Therefore, a **programming language** is particularly suitable for building automatic learning models for predicting the positivity or negativity of a text [1]. With the advent of Data Science, two languages have been established: Python and R. Both have a number of libraries for the processing of natural language and lexical resources of different kinds. NLTK is the Python library, while for R the module is TM which has native packets for reading different PDF and XML file formats.

If you are unfamiliar with the programming languages, another option is to rely on **tools with interfaces** that allow you to build the necessary processes like Knime, Weka, Rapid, Amazon Comprehend [2], Google Cloud [3], Stanford's CoreNLP sentiment analyzer [4].

In this project we are going to choose programming approach to achieve sentiment analysis and text summarization, here we are choosing python language to do sentiment analysis and text summarization because it has multiple methods, libraries and tools so with the help of that we can automate our code to easily get sentiment analysis and text summarization.

5.2 Method and algorithms used for sentiment analysis

A. Data preprocessing

a) Convert Paragraphs to Sentences:

First we have to convert a whole paragraph into sentences. Most easy way to converting a paragraph to sentence is to split a paragraph whenever a period is encountered.

b) Text Preprocessing:

Once we convert paragraph to sentences we have to remove all the special characters, stop words and numbers from all the sentences.

c) Tokenizing the Sentences:

After that we need to tokenize each sentences to get each word that exist in our paragraph .After tokenizing the sentences**.**

For data preprocessing NLP tools which is NLTK library and their different functions like stop words, string, tokenization to preprocess our data and make it able to fit in our ML model.

B. ML and NLP algorithms

Here we are using lexicon-based approaches, a sentiment is defined by its semantic orientation and the intensity of each word in the sentence. For this we require pre-defined dictionary which able to classifying negative and positive words. Generally, a text message will be represented by bag of words.

Once we assign scores to all individual words we have to find final sentiment and for that we use some pooling operation like taking average of all sentences. Here TextBlob is used for sentiment analysis, TextBlob is a python library for Natural Language Processing (NLP). TextBlob use NLTK to achieve its task.

NLTK is python library which gives us an easy way to access lot of lexical resources and allows users to work with categorization, classification and many other tasks. TextBlob is a simple library which supports operations on textual data and complex analysis.

TextBlob returns two function one is **polarity** and second is **subjectivity**. Polarity lies between -1 and 1,where -1 defines a negative sentiment and 1 defines a positive sentiment. Negation words reverse the polarity. Subjectivity lies between [0,1]. Subjectivity define the amount of factual information and personal opinion contained in sentence, if it defines higher subjectivity it means sentence contain more personal opinion than factual information. So by using TextBlob library and its polarity function we are able to find the sentiment of each tweet as positive or negative and

combining all tweets together we can easily find that how many percent of tweets are positive, negative or neutral.

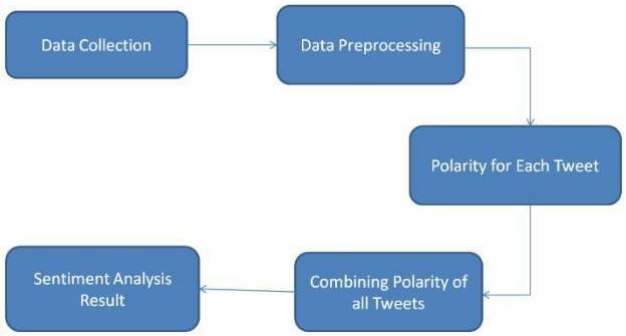

Fig 2 – Sentiment Analysis Workflow diagram

6. Text summarization

Text summarization is a technique of shortening long pieces of text. The main purpose is to create coherent and fluent summary having only the main points outlined in the document. Automatic text summarization becomes a common problem in machine learning and natural language processing (NLP). Machine learning models are usually trained to understand documents and distill the useful information before outputting the required summarized tests.

Algorithm used in text summarization

In our project we have used same data preprocessing techniques for text summarization and sentiment analysis by using some natural language processing kits function we can easily do that. Here we are using extractive based text summarization approach and TextRanking algorithm to summarize our tweets.

After preprocessing we are going to Find Weighted Frequency of Occurrence. To find the frequency of occurrence of each word, we use a variable. This variable does not contain any punctuation, digit or special character so we use this variable to find frequency of occurrences, after that we store all the stop words in stop word variable from NLTK library and then we loop through all the sentence and check whether the variable is stop word or not . If not, we proceed to check whether the words exist in a dictionary or not. If the word is occurred first time then we add that word to the dictionary as a key and set its value as 1. Otherwise, if that word already exists in the dictionary then its value is updated by 1.

Finally, to find the weighted frequency, we can simply divide the number of occurrences of all the words by the frequency of the most occurring word.

A. Calculating Sentence Scores:

Now we have calculate weight frequency for all the words in our paragraph. After that we have to find score for each individual sentence by adding weighted frequency of each word that occurred in that particular sentence.
We first create an empty a dictionary. In this dictionary key will be the sentence and value will be there corresponding scores. Next, we loop through each sentence in a list and tokenize the sentence into words. We then check if the word exists in a dictionary. This check is performed since we created a list from the object; on the other hand, the word frequencies were calculated using an object, which doesn't contain any stop words, numbers, etc. In our summary we don't want any long sentence hence we are using only those sentence which have less than 30 words in it .Next, we check whether the sentence exists in a dictionary or not. In our dictionary if the sentence does not exist then we add that sentence as a key and add its values as weighted frequency of first word in sentence, otherwise if the sentence is already exist in dictionary then we add weighted frequency of word to existing value.

B. Getting the Summary:

Now we have a dictionary that contains sentences with their corresponding score. For summarizing our paragraph we can take N sentence which have highest score.

Here we are using heapq library to find N sentence with highest score and for that we are using nlargest function in heapq library to find 7 sentence with highest score.
By using this algorithm, we are able to find summary of our tweets.

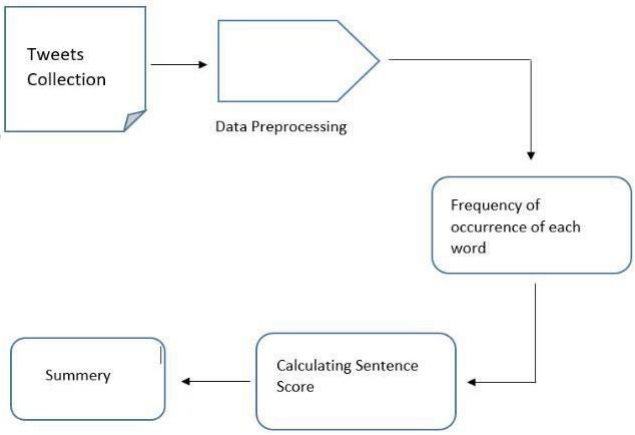

Fig 3 – Text Summarization Workflow diagram

8. CONCLUSION AND FUTURE SCOPE

Sentiment analysis and text summarization over the Twitter data continues to very interesting and helpful task. This project would help proving the clear and unbiased look at the topic for the users thus preventing the misguided and fake reports. The simple UI would make users the access data and analysis results without any difficulty thus satisfying the main objective of
the project. It would also provide the polarity of the given topic making it easy to classify and filter the topics based on positive, negative or
neutral sentiment. For the Future scope this project can be used for various other topics or can be in real time on social media platforms to prevent misguiding of the users for controversial topics. The project can also be embedded with the support the multiple languages and the use of OCR technology for analyzing data through various media sources like Images and Memes as it one of the most effective and widely used method for stating ones opinion on a wide scope.

9. References

1. https://www.extrasys.it/en/smartblog/sentiment- analysis-how-to-describe-your-products-positively
2. https://aws.amazon.com/getting-started/hands-on/analyze-sentiment-comprehend/
3. https://cloud.google.com/natural-language/docs/analyzing- sentiment
4. https://stanfordnlp.github.io/CoreNLP/sentiment.html
5. https://monkeylearn.com/sentiment-analysis/
6. https://towardsdatascience.com/a-quick-introduction-to- text-summarization-in-machine-learning-3d27ccf18a9f
7. https://www.extrasys.it/en/smartblog/technologies- used-for-data-mining-sentiment-analysis
8. https://www.ijcaonline.org/research/volume125/number3/dandrea-2015-ijca-905866.pdf
9. https://www.aiperspectives.com/twitter-sentiment-analysis/
10. https://www.sciencedirect.com/science/article/pii/S209 0447914000550#t0005
11. https://stackabuse.com/text-summarization-with-nltk-in-python/
12. https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524#:~:text=TextBlob%20is%20a%20python
13. %20library%20for%20Natural%20Language%20Processing%20(NLP).&text=After%20assigning%20individual%20scores%20to,and%20subjectivity%20of%20a%20sentence.
14. https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/
15. https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/