Movie Review Comparison

Ankita Gadge, Mrunal Dharmendra Maniar, Nikhil Yathindra, Swapnil Borse

***Abstract*—The project aims to provide a statistical description of the sentiments obtained for a particular movie from different data sources. The user is required to provide the name of the movie, the official links of the trailers on Facebook and YouTube. The comments on these trailers would be procured using the official API provided by Facebook and YouTube. Using NLP we would obtain a brief overview of the percentage of positive and negative reactions to the trailer. This would be compared to the ratings obtained from APIs provided by OMDB and TMDB. The obtained statistics would be portrayed graphically to the user using Bokeh library.**

I. Introduction

THIS project would help in providing a brief analytical idea about the performance or the box office collection of a movie based on the response obtained to the trailer. The response mainly considers the comments obtained on the trailer and subjecting them to sentiment analysis using Natural Language Programming(NLP). For the scope of this project, the data channels considered are Facebook, YouTube, open movie database (OMDB) and TMDB.

The reason for this selection is that Facebook and YouTube have an average of million users everyday who interact with the subject being considered here. They also have a reliable data scraping mechanism that provides reliable information about user interactions with movies and especially with the trailers of these movies. They also provide secured access to developers without compromising the privacy of their users. OMDB and TMDB provide a very reliable rating score for movies.

This project mainly targets anyone who wishes to study, compare and analyze public sentiments about a movie on these major social media platforms. One can get a comprehensive idea about how well a particular movie is being received throughout the social media world. One may use these sentiments to model strategies, devise decision plans and essentially try to enhance the business.

The second section of this paper provides detailed description of data extraction from social media sources. Third section of this paper provides an idea of the sentiment analysis and describing the response graphically.

II. Data Extraction

The main idea of this system is to procure data from Facebook and YouTube and comparing the response to that obtained from OMDB and TMDB.

API or application programming interface is a set of subroutines or protocols using which a communication channel can be established among the software components. Most of the social networking websites make their APIs external using which developers can access data from their data store. But this access is provided in such a way that the privacy of user data on social network is not compromised. This security is provided through access tokens or secured keys. This type of access is generally termed as OAuth or OAuth2 access. In order to obtain this access, most of the social networking websites require the application developer to register their application before hand to obtain their access tokens to make API requests. Most of these API requests are carried out in terms of messages governed by Hyper Text Transfer Protocol (HTTP) or secured version of HTTP which is abbreviated as HTTPS. Another routine used in APIs for this project is Representation State Transfer (REST) format of communication using API. Using REST APIs, we make a GET request to obtain information, POST request to upload new information, PUT request to modify information and DELETE request to remove information. Success or failure of this request is indicated using status codes.

For this project, all the response obtained from API is in JavaScript Object Notation (JSON) format which can be easily read and formated for display. JSON is a lightweight data-interchange format in which information is stored in terms of key-value pair.

A. Facebook Data Extraction :

Facebook data extraction is done through means of graph API which is provided by Facebook. Graph API [1] is the primary medium through which data is added or obtained from the facebook social graph externally. As the name suggests the, facebook social graph is a graph in which the nodes are entities like user profiles, photos, video, comments, etc. The edges of this graph identify the association. For example, if the picture of a user named X is identified by node Y, there would be an edge between node X and node Y. Fields provide details of the node like the name of the user. In the version of API used for this project, Facebook permits an application to make 200 API requests per user per day. So if an application has 100 users, the application can make 200,000 API requests to the Facebook graph. This is the limit constraint implemented by Facebook. However, it is not the same for requests made for Facebook pages or Facebook ads which is beyond the scope of this project.

B. YouTube Data Extraction :

Google provides an API to extract comments from videos uploaded on YouTube [4]. The same API provides the options to subscribe to a channel in YouTube, create playlists,upload and delete videos of a given user, manage playlists, add captions to videos , etc. For the scope of this project, we are using the YouTube comment threads service endpoint which is made available to developers by means of access tokens.

This endpoint has a method to procure the list of comments for a given video uploaded on YouTube. For the sake of this project we are fetching the link of trailer on YouTube from TMDB.Google calculates the usage limit for YouTube API by assigning a cost to each API operation requested. For instance, a simple read operation to obtain the id of video has a cost of 1 unit, where as an API request that performs the operation of uploading a video is approximately 1600 units. On similar basis, each API request has a cost associated with the requested operation. Based on the type of access requested from YouTube, the rate limit for particular number of API operations in a day is provided by Google. Google also provide a dashboard which would assist the application developer to monitor the number of API requests made by the application in a day.

C. TMDB and OMDB data extraction :

TMDB or The Movie Database [2] holds information of almost all the movies and their details like the cast, release date, their social media handles,etc. It also provides details of Top 10 movies in a given category. It also has similar details for TV shows. TMDB API is made available to developers by means of a unique identification key without which no service request can be made to TMDB. OMDB [3] or the Open Movie Database also operates on similar principles. Both of these APIs require the movie name to provide all the details of that movie. For this project, we are using this API to obtain the ratings of this movie. TMDB enforces a rate limit that permits 40 API requests per second which is identified by the IP address of the request made and not the API token or key. At the same time, OMDB API enforces a rate limit that permits only 20 concurrent connections.

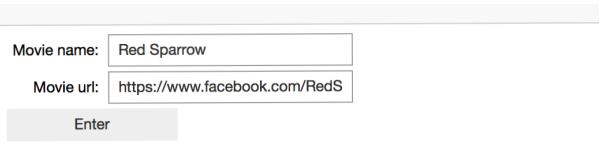

Fig. 1. Illustration of User interface

III. Sentiment analysis and Natural Language Processing (NLP)

Natural-language processing (NLP) is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to fruitfully process large amounts of natural language data. NLP has variety of applications. Few of them being machine translation, automatic summarization, sentiment analysis, text classification, question answering, etc. NLP deeply relies on machine learning algorithms according to the paradigm for which it is being used. For instance we are using NLP for sentiment analysis of comments. Sentiment analysis, the ex- sentiment analysis traction of sentiment, the positive or negative orientation that a writer expresses toward some object. A review of a movie in our case, book, or product on the web expresses the author's sentiment toward the product, while an editorial or political text expresses sentiment toward a candidate or political action. Automatically extracting consumer sentiment is important for marketing of any sort of product, while measuring public sentiment is important for politics and for market prediction. The simplest version of sentiment analysis is a binary classification task, and the words of the review provide excellent cues.[5] the algorithm would classify the comment into three categories namely : positive, negative and neutral.

A. Reason for selecting Naive Bayes Classifier (NBC)

Naive Bayes classifiers are a family of simple "probabilistic classifiers "based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

Use of NBC involves doing certain number of counts for making a classification. In case of NBC, one may assume that the features used would be independent from one another. For example, if we assume the actor of the movie to be a feature for training the model and the location where the movie was filmed to be another feature, we know that these features are not interrelated. Assuming conditional independence, a Naive Bayes classifier will work quicker than other discriminative models like logistic regression, so we need less data for training the model. Its main disadvantage is that it can't learn interactions between features (e.g., it can't learn that although you love movies with Brad Pitt and Tom Cruise, you hate movies where they're together).

For the scope of this project we are using Naive Bayes Classifier(NBC) to classify comments as positive, negative or neutral. In this project, we have used python textblob library to perform sentiment analysis on the movie reviews and movie comments. TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. The library internally uses algorithms like Decision Trees, Naive Bayes classifier to classify the input stream of data. By default, the Naive Bayes Classifier uses a simple feature extractor that indicates which words in the training set are contained in a document. One can override this feature extractor by writing one's own. A feature extractor is simply a function with document (the text to extract features from) as the first argument. The function may include a second argument the training dataset, if necessary. TextBlob is written on top of NLTK which uses Naive Bayes Classifier to polarize sentences based on the comments.

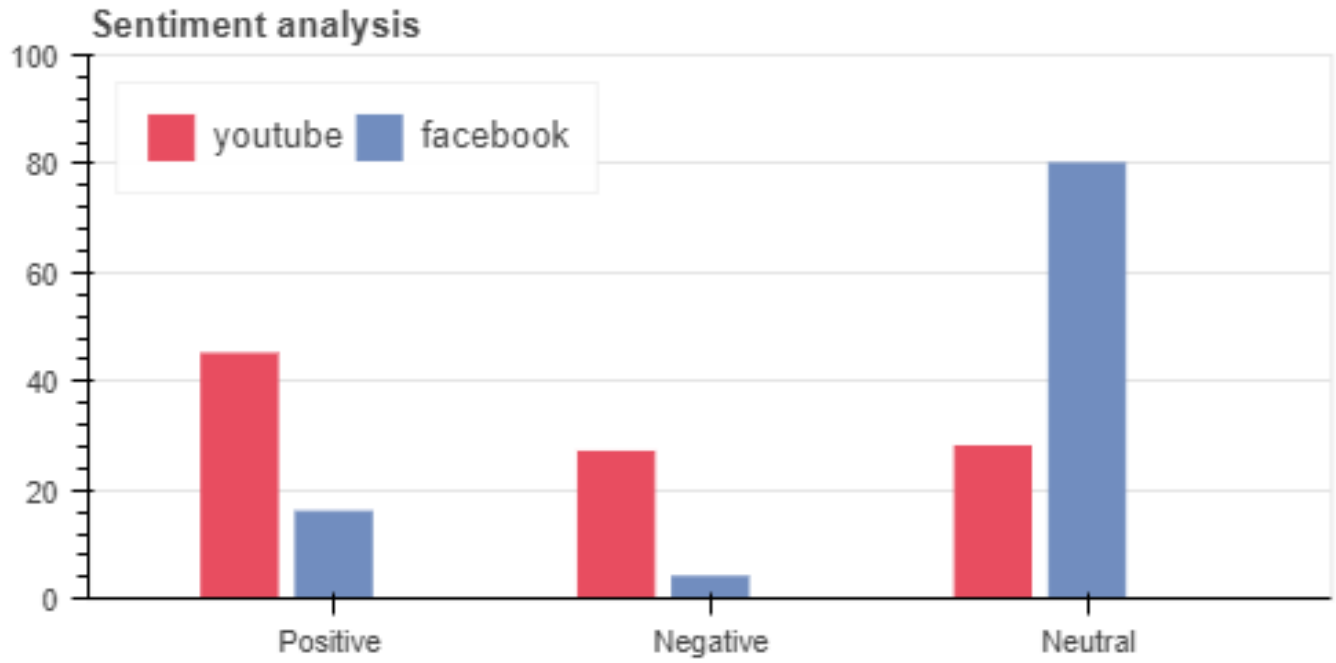

Fig. 2. Graph illustrating Facebook and YouTube results

IV. Methodology

The working of this project is simple and straight forward. Working is as follows :

- The user is required to provide the name of the movie and the link of the facebook trailer in the user interface as illustrated in figure 1.
- From the link of Facebook trailer, extract the video id on Facebook and obtain all the comments of this video using the Graph API. These comments are subjected to sentiment analysis to obtain a classification of these comments as positive, negative and neutral. A graph is plotted for the same as illustrated in Figure 2.
- On the same grounds, using YouTube Data API, we fetch all the comments or reviews for the official trailer on YouTube and subject these comments to sentiment analysis. This graph is combined with one obtained for Facebook. As observed from the graph in figure 1, we see that the graph contains a combination of both YouTube and Facebook comments.
- For the sake of comparison, obtain the ratings for the movie from TMDB and OMDB APIs. The graph for this rating is illustrated in Figure 3.
- The next phase to combine the comments and calculate the overall percentage of positive, neutral and negative comments. This is illustrated in Figure 4.

  The graphs shown here were plotted using Python's Bokeh library. Bokeh is an interactive visualization library that targets modern web browsers for presentation.

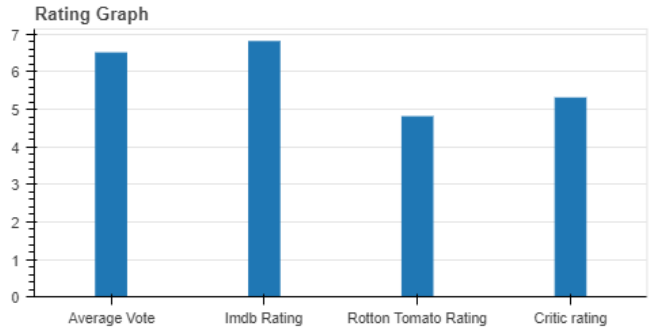

Fig. 3. Graph for ratings from obtained using IMDB and OMDB API

Its goal is to provide elegant, concise construction of versatile graphics, and to extend this capability with high-performance interactivity over very large or streaming datasets. Bokeh can help anyone who would like to quickly and easily create interactive plots, dashboards, and data applications. As observed from the two graphs in Fig.1 and Fig.2, we observe that the values obtained from Facebook and YouTube are fairly close to that obtained in the ratings from IMDB and OMDB API.

V. Calculation using Naive Bayes Classifier

Although, we use the inbuilt library function provided to us by Textblob, this section provide a brief overview of how classification is done using NBC.[5]

If we assume that we are classifying results to a class C, we first identify N_C which signifies the number of data elements that belong to the class C in the training data. Similarly, we identify N_T which signifies the total number of data elements in the training data. Thus, we are required to calculate P(C)

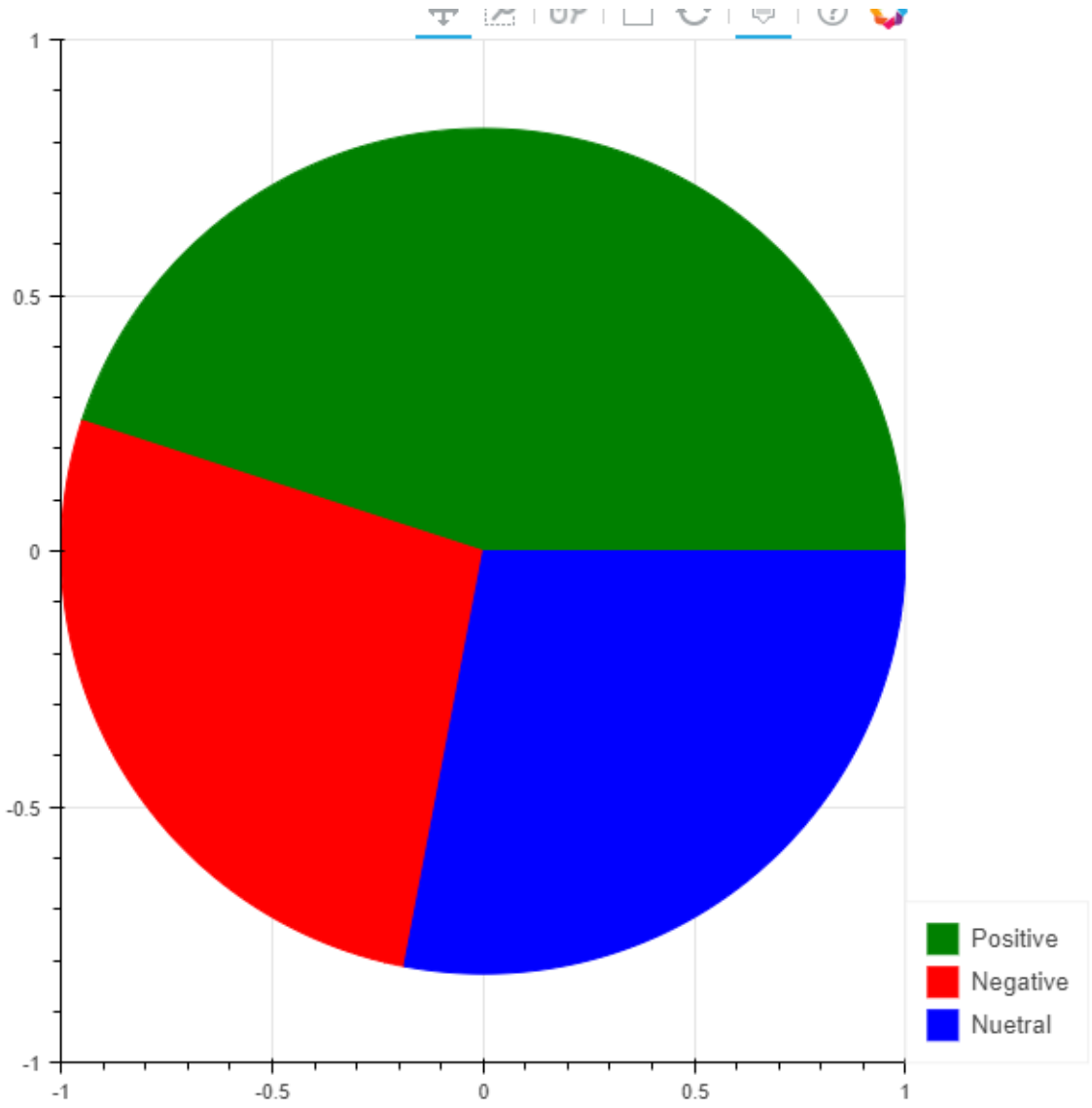

Fig. 4. Overall percentage of positive, neutral and negative comments

which is given by N_C/N_T.

In the vocabulary, one may assume that there are words that belong to different classes and not just class C. So we group all the data elements that belong to one big data element. Then we use the frequency of a word w_i in t this grouped data element to give a maximum likelihood estimate of the probability which is given by :

$$P(w_i \mid C) = \frac{P(C|w_i) * P(w_i)}{P(C)} \quad (1)$$

where $P(w_i \mid C)$ gives us the probability of w_i belonging to class C and $P(C \mid w_i)$ is the likelihood of w_i to belong in class C. This way probability is calculated for all data elements for words and the model is trained. After the model is trained, for any input comment, for each word in the comment, we see which class a word has been defined to belong and the cumulative probability of words with respect to their class would give us the class to which the comment belongs to.

VI. Results and Conclusions

As mentioned before the sentiment analysis module of this project draws its roots from NLTK. The results obtained from sentiment analysis was very convincing. The percentage of positive, negative and neutral comments was quite close to the ratings obtained from TMDB and OMDB. This means that the success or failure of a movie can be projected by using the reviews the trailer has got on Facebook and YouTube.

VII. Future Enhancements

This project can be further enhanced to incorporate comments from various other social media channels like Twitter, Pinterest, Snapchat, Tumblr, etc.

The classification of comments can be enhanced by training a neural network that would provide the classifier the ability to identify sarcastic comments and provide appropriate classification.

The scope of this project is confined to the official trailers of a movie. This can be improvised by considering the fan pages and fan made trailers. This would cover a larger group

of audience and provide itemized classification like which age group of people enjoy what genre (action,comedy,drama,etc) of movies.

Further, we can also provide information for the feedback of the trailers based on location as most of the social networking channels support providing comments broken down on the basis of location, timeframe,etc .

REFERENCES

[1] https://developers.facebook.com/docs
[2] https://www.themoviedb.org/documentation/api
[3] http://www.omdbapi.com/
[4] https://developers.google.com/youtube/
[5] Daniel Jurafsky James H. Martin, Speech and Language Processing (Naive Bayes and Sentiment Classification)