“YouTube video feedback analysis using Natural Language Processing”

Md. Anisur Rahman Rony
Dept. of CSE
Daffodil International University
Dhaka, Bangladesh
anisur15-7880@diu.edu.bd

Meherin Amir
Dept. of CSE
Daffodil International University
Dhaka, Bangladesh
meherin15-7922@diu.edu.bd

Nujhat Tabassum Amithy
Dept. of CSE
Daffodil International University
Dhaka, Bangladesh
nujhat15-7750@diu.edu.bd

Shahariar Rabby
Dept. of CSE
Daffodil International University
shahariarrabby@gmail.com

***Abstract*— Now a day’s data is increasing rapidly around the world. For this reason, human can’t handle it manually. And so we need the help of computer. For example, I have a YouTube channel and I have huge number of subscribers and followers who regularly watch my videos which are uploaded on my channel. If there are huge number of comments, I can’t read every comment due to lack of time. And so I will be unable to understand that which comments are positive and which are negative. If I don’t know whether most of the feedback is negative or positive, it will be difficult to improve my video quality. That’s why review or feedback analysis is very much important. There are several Machine Learning and Natural Language Processing algorithm to solve this problem and we are going to implement these in our project. Moreover, we will connect it to the web by deploying our model into the web. In this case we will use Python Flask, Web Scraper etc. For collecting primary data of YouTube video (negative/positive feedback), we have used web Scraper. And data been finally collected from that as a csv format to train our model. And then we will use this data to train our model by converting into tsv format. Our final outcome will be web based review analysis app.**

***Keywords*— Natural Language Processing, Comment Analysis, Web Scraper, Selenium, Python Flask, Random Forest Classification.**

I. Introduction

YouTube is one of the greatest platforms for video where huge amount videos with many categories are uploaded and seen by the millions of people. There are many educational and entertainment related videos exists. There are millions of videos and millions of comments available. Among those comments if we want to identify negative or positive comments it will be very much difficult and our valuable time will be lost through this task. But Machine Learning and Natural Language Processing has made it easier. But applying these we can easily track it. Among thousands and millions of comments it is identified smoothly which one is positive and which one is negative. Our feedback analysis process can be easily handled by it. And one question may be raised that why we need to do feedback analysis of YouTube Videos. First of all, suppose I am searching any education video in the YouTube. Then we will find huge number of videos. Among those if we want to find which one essential, we can easily do that by focusing one comment section delivered by the previous people. If the video is good, then definitely people will post positive comment. And if the video is not good or not helpful people will make a negative comment. And by analyzing YouTube comment feedback we can easily identify which one video may be helpful for me. If the maximum comment is positive, then we can ensure that the video may be helpful for me regarding our study or any purpose. Similar things can be happened in other context such as which video is more entertain able or anything. It will make us understandable before watching any videos that which video may be skipped without watching it from the ratio of positive and negative comment. If there are so many negative comments the we can estimate that the video may be useless or it may contain negative things such as hate speech, useless or harmful things. Then we can simply avoid it save our valuable times from wasting it. And this is from the side of viewers. If we consider from the side of uploaders of the videos comment feedback analysis will also play a great role. Suppose someone uploaded a video on YouTube. Then there he/she saw that millions of comments are there. And he/she wanted to know that which comments are positive and which comments are negative. Then it will be impossible for him/her track it manually by reading comments one by one. He/she can solve this by applying feedback analysis technology by solving this problem. If there are maximum negative comments and few positive comments, he/she can improve the video quality with good contents. Then it will be the part of self judgement process for improving video quality. From the above both perspective feedback analysis of YouTube comments is very much important. In this paper we will show the process of tracking positive and negative YouTube comments automatically using Natural Language Processing, Python Flask Framework, Web Scrapper etc. The process will be shown step by step which will be easy to understand.

II. WORKING OVERVIEW

To complete our research, we have to maintain some activities sequentially and step by step. Our total working flow-chart diagram is given below and all the steps will be explained in details. We have divided our whole process into several steps. Such as Making a Web Scraper, Collecting Dataset, Data Processing, Making Model and finally deploying our model. In this case, we kept two options. One is deploying our model to Web. In this case we will use python flask. And another option is deploying our model with Web Scraper. And in this case using a web scraper we can automatically track the YouTube comments which are positive or negative. Now we are going to explain the whole process in details and in step by step.

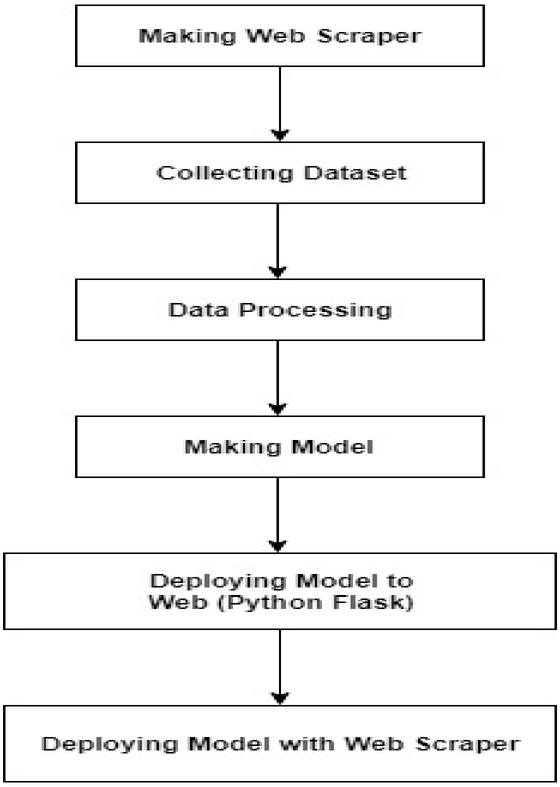
Making Web Scraper
Collecting Dataset
Data Processing
Making Model
Deploying Model to Web (Python Flask)
Deploying Model with Web Scraper

Figure 1: Working Overview.

A. Making Web Scrapper

Web Scraper is the which scrap the necessary information from the website. In this case we need a web scraper for multiple purpose. First of all, we can collect our necessary data by using web scraper which is useful for training our model. Moreover, if we want to track the positive/negative feedback instantly in online that means if we want to deploy our model to online and see the result instantly we can use Web Scraper. So, at first, we need to make a web scraper by which we can do these activities. There are many types web scraper for multiple purpose. In our project we have to use it for collecting YouTube comments and tracking its status that means whether it is negative or positive. As we want to deploy it in python-based web framework and our model will be implemented through python libraries, we have chosen python related web scraper. For making Web scraper, we can use beautiful soup, selenium etc. In this case we used selenium web scraper. Because it is more efficient than others to me. Selenium can scrap the comments very well even if while it is scrolling. And by scrolling it our comments will be tracked efficiently. In this case chrome driver was also used. In our web scraper YouTube video link will be taken as input and YouTube comments will be its output. In this case only English comments will be tracked. Emoji's will also can be identified by it which is very much helpful for our model.

B. Collecting Dataset

After making web scraper next work is to collect data. In this case several methods are following in collecting data. First of all, web scraper was used to collect data as a csv format. Then among those data negative and positive comments were tracked manually as 0 and 1. Moreover many labelled sentences are collected as dataset which are either negative or positive. As we know this labelled sentence are similar to negative and positive comments, we used these as our dataset. Different types of review dataset such as movie review, restaurant review dataset was also merged in this case. Finally, after collecting dataset as csv format we convert it into tsv format for making our code easier. Finally, we have collected almost three thousand datasets for building our model.

C. Data Preprocessing

After collecting the dataset our next step is to process data. We divided or whole data processing steps into several steps. Such as using regular expression, removing unnecessary symbols and numbers, lowering texts, splitting text, taking root word. Removing stopword, making bag of words etc.

First of all, we used regular expression to take the exact data which is matched with the regular expression. Such as taking data which are from English character A-Z and a-z. Rest of the symbols will be removed from that data. This work is done using a library called re which are imported from python library. Thus, unnecessary symbols and numbers are also removed from the data.

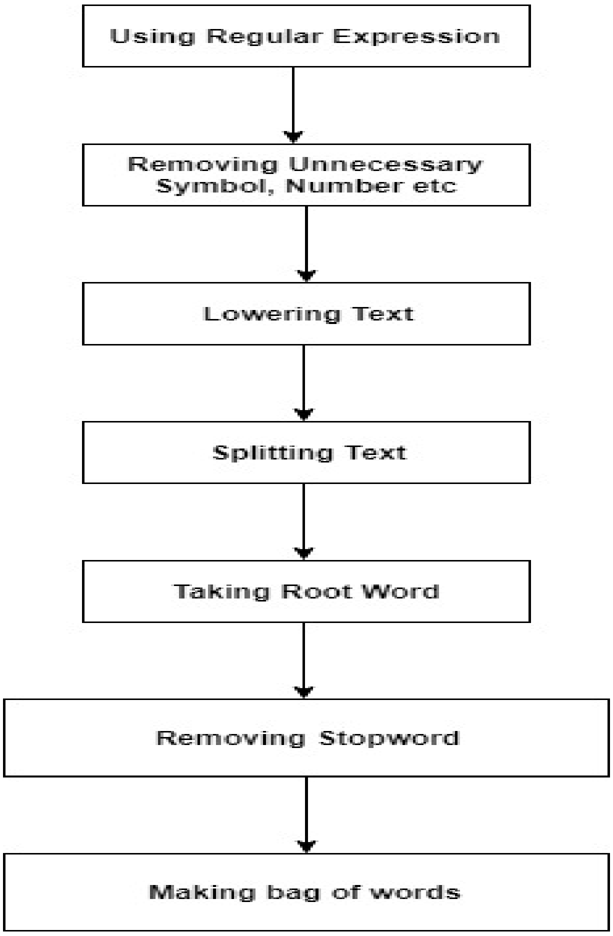

Figure2: Processing Data.

Next step is to convert the whole character to lowercase. As there are not differences between the upper- and lower-case character according to the meaning of sentences. Moreover, if we use lowercase it would be helpful for our code in building our model. This job is done by using a method called lower.

After that, datas are splitted using split method.

Suppose that there is a sentence called “I am reading a book”. In this sentence reading contains extra “ing” is not necessary in evaluating the comments as positive or negative. We need the main root word because in different sentences this word can be found in different format such as read. So, we need the root word or main word. The process is done via PorterStemmer. This process is called stemming.

There are many words in sentence which existence does not make any meaning and which should be removed or ignored to avoid complexity and to get better performance. These are called stop words. Such as a, an, the, in etc. This work is done via stopwords which is imported from nltk.corpus.

Finally, these are joined to an array.

Last step of data processing is bag. And this bag contains words. That means bag of words. This work is done using sparse matrix. We use CountVectorizer class for this work. In this case only those words are taken not more than one times.

D. Making the Model

Completing the work of data processing next work is to make model. In this case, we used random forest classifier algorithm to make our model as it works better than other algorithms. After processing data, we splitted our dataset into test set and train set. In this case we kept 80% data as train set and 20% data as test set for better accuracy. Splitting dataset was done using train_test_split which was imported from sklearn.cross_validation. After that our final model creation process begins. At first, we imported RandomForestClassifier from sklearn.ensemble. Then an object of it is created. To make the object we have three parameters which we can use inside the RandomForestClassifier. They are n_estimators, criterion, min_samples_split etc. Default value of n_estimators is 10. We can also use different values of those parameters. Random forest algorithm is implemented in different ways to different framework. In Scikit-learn it is it finds the node importance by Gini Importance guessing two child nodes only. The equation is given below:

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

Where,

ni sub(j) = the importance of node j

w sub(j) = weighted number of samples reaching node j

C sub(j) = the impurity value of node j

left(j) = child node from left split on node j

right(j) = child node from right split on node j

Here calculation of importance of feature is done as:

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k\in all\ nodes} ni_k}$$

Where,

fi sub(i) = the importance of feature i

ni sub(j) = the importance of node j

Then normalization is occurred from the range 0 to 1 by dividing sum of feature importance values which were calculated above and the equation is:

$$normfi_i = \frac{fi_i}{\sum_{j \in all\, features} fi_j}$$

And finally, the Random Forest feature importance is calculated as the average of all trees. The equation is given below:

$$RFfi_i = \frac{\sum_{j \in all\, trees} normfi_{ij}}{T}$$

Where,

RFfi sub(i) = the importance of feature i calculated from all trees in the Random Forest model

normfi sub(ij) = the normalized feature importance for i in tree j

T = total number of trees

And it is all about our classification model. And by this model we had trained our dataset so that it can predict about the new data that means new YouTube comments efficiently.

E. Deploying Model to Web Using Python Flask

Our next work is to deploy our model to web. In this case we used python flask. We used python in this case as we developed our model using python library and it is easier to implement machine learning model in python and so python-based framework flask.

In this case, we created two pages using html, css etc. One web page for taking input from the text area of the YouTube comment. Another web page for showing the result. In this case, if the possibility of the comment is negative then it will show it as a positive comment. On the other hand, if the comment is negative it will show as a negative comment.

At first in app.py folder inside predict function our model which is mentioned has been created. Then using pickle library, it has been saved. And then our model is given to predict for new comment. As we classified our dataset as 1 and 0 as positive and negative comment, we have used a condition to predict for our model. If the prediction for new comment is equal to 0 then it is negative comment and if it is equal to 1 then it will predict as a positive comment. Thus our model is used to predict YouTube comment feedback in the web using Python Flask.

F. Deploying Model with Web Scraper

In the web flask we can see that if we want to analyze the YouTube video comments we had to put comments in the text area manually. It is not the efficient way. So it is great problem if there are huge number of comments. We can solve this problem by a method. That is automatically tracking the negative and positive comments. In this case we can use web scraper. Web Scraper scrap the data to extract. It is done using HTTP protocol and it is done in the web browser. It is the common work of web scraper. For YouTube comments we can use it to scrap the YouTube comments. And it will be done automatically. Not only scraping YouTube comments but also it will identify the negativity or positivity of the comments if we attach our model with it.

We used selenium webdriver to make our YouTube comment scraper. Though there are many other options to make web scraper, we used selenium because it is more efficient and easier to implement with python. There is also functionality of detecting emoji's. Our web scraper can take the data of video title, author or user of comment and the main comment which will be predicted using our model automatically. It works only with English language. And it will not work on other languages. It will take the link of YouTube video as input and then it will take the comments of that video by scrolling one by one within a particular time. At the same time, it will predict the feedback of those comments and will print negative or positive after that comment. Moreover, it will also show the total number of comments which were detected by the scraper, total number of positive comments, total number of negative comments and the percentage of them. After the YouTube link input, a chrome driver will be automatically opened with the video. And after doing all the above works chrome driver will be closed using close function.

III. EXPERIMENT AND OUTPUT

We have trained our model with almost 3000 of dataset. Among them 80% were kept as training dataset and 20% were kept as test dataset.

In this situation our model gave almost 80% accuracy. If we train our model with more and huge number of dataset, then we hope that it will give more accuracy. So we should increase the number of dataset for more accuracy and to run our model efficiently.

IV. CONCLUSION AND FUTURE WORK

In this paper, we classified YouTube comments into two classifications. One is positive and another is negative. Our dataset size is almost 3000. And its accuracy is almost 80%.

In our project there are some limitations. Such as it is only based on English Language. It can't give any result with other languages. Moreover, It's accuracy is not so much good which should be developed.

So, in future, we have to overcome our limitations. In this case many steps can be taken in future. At first we can apply it with the others languages along with English. We can train our model with huge number of dataset so that our accuracy may be increased. There are not only two categories in YouTube comments, we should increase the classification number. We can classify it with many categories. In this case our efficiency will also be increased and our model will give right input.

Moreover, we can use others algorithm and compare those output with our current model output. We can also use our customized algorithm to check if it gives better output.

ACKNOWLEDGMENT

We are thankful to DIU Natural Language Programming and Artificial Intelligence Lab for giving us the necessary equipment regarding our research. Also, we would like to give thanks to our course teacher for guiding us and showing us the right direction to our research. Without his support it might be difficult to continue our research.

REFERENCES

[1] S. Redhu, S. Srivastava, B. Bansal and G. Gupta, "Sentiment Analysis Using Text Mining: A Review", International Journal on Data Science and Technology, vol. 4, no. 2, p. 49, 2018. Available: 10.11648/j.ijdst.20180402.12 [Accessed 16 August 2019].

[2] S. Manna, "Sentiment Analysis based on Different Machine Learning Algorithms", International Journal of Computer Sciences and Engineering, vol. 6, no. 6, pp. 1116-1120, 2018. Available: 10.26438/ijcse/v6i6.11161120 [Accessed 17 August 2019].

[3] S. Muthukumaran and D. P.Suresh, "Text Analysis for Product Reviews for Sentiment Analysis using NLP Methods", International Journal of Engineering Trends and Technology, vol. 47, no. 8, pp. 474-480, 2017. Available: 10.14445/22315381/ijett-v47p278 [Accessed 16 August 2019].

[4] D. Mohey, H. M.O. and O. Ismael, "Online Paper Review Analysis", International Journal of Advanced Computer Science and Applications, vol. 6, no. 9, 2015. Available: 10.14569/ijacsa.2015.060930 [Accessed 17 August 2019].

[5] A. Hasan, S. Moin, A. Karim and S. Shamshirband, "Machine Learning-Based Sentiment Analysis for Twitter Accounts", Mathematical and Computational Applications, vol. 23, no. 1, p. 11, 2018. Available: 10.3390/mca23010011 [Accessed 17 August 2019].

[6] D. Kawade and D. Oza, "Sentiment Analysis: Machine Learning Approach", International Journal of Engineering and Technology, vol. 9, no. 3, pp. 2183-2186, 2017. Available: 10.21817/ijet/2017/v9i3/1709030151 [Accessed 17 August 2019].

[7] S. Shayaa et al., "Sentiment Analysis of Big Data: Methods, Applications, and Open Challenges", IEEE Access, vol. 6, pp. 37807-37827, 2018. Available: 10.1109/access.2018.2851311 [Accessed 17 August 2019].