YouTube Title Prediction Using Sentiment Analysis

Yajat Uppal, Arnav Khandelwal, Jaideep Singh Chahal

Computer Science Engineering Bennett University, Greater Noida, India

---***---

***Abstract*—The Current Situation Social Media sites like YouTube, Facebook and Instagram control the world in this age. They let us connect, share, and consume all kinds of content from across the world with minimal efforts. This feeling of connection, in a way that is bigger than ever imagined before comes with its own quirks. The upsides always come with some downsides. While social media sites easily allow you to connect with others, they also provide anonymity along with a depraved notion of not having to face consequences for our actions. Anonymity while being essential to maintain privacy is seen as a source of expressing hostility to each other. There are all kinds of toxic comments on social media including spam, abusive words, racist remarks, and more.**

Moreover, feeling human emotions like sympathy without knowing or being able to see the other person requires mindful- ness that many users lack. YouTube is the largest video sharing and watching platform on the internet. Millions of users log on every day to create or watch videos. However, it is difficult for content creators to interact with users. As mentioned above, comments are flooded with spam, hatred, and toxicity. Therefore, it is hard to find genuine and constructive feedback for content creators. The Social Media domain is massive making it hard to comprehend. Therefore, filtering and classifying comments provides us with a much need edge in this ever-expanding world.

***Index Terms*—Deep Learning, Transform Neural Network, Regression, Classification, Logistic Regression**

I. INTRODUCTION

Our project is a sentiment analysis algorithm that analyses the sentiments of comments using Linear Regression. Our algorithm evaluates the comments of a YouTube video and judges whether they are positive or negative. The results generated using this analysis are showcased using Graphs. Various parameters are compared to disclose useful and relevant conclusions drawn from the graphs. Furthermore, this algorithm generates new titles for a YouTube Creator. The videos of a YouTuber are ranked on the basis of the number of positive/negative comments. The titles of videos with an overall positive response are scraped. Keywords are extracted from the titles. The algorithm then searches these keywords on YouTube to find videos by other Creators on the topic. The titles of these videos are used to generate a new title for the topic using a Deep Neural Network.

The first part of our program is to analyse and categorise the videos based on comments. From a given YouTube channel link/ID, the program visits and scrapes all videos available on the channel. It then extracts the comments of the video and analyses whether they are positive or negative. The extracted comments are judged and categorised with the help of a machine learning based NLTK (Natural Language Tool Kit), which contains various text processing libraries for python. It also detects spam or bot comments and removes them. Our algorithm scrapes comments from YouTube using the YouTube Data API. The classified comments are then later used to judge the video and put them in their corresponding category.

Once the comments are analysed and categorized using Sentiment Analysis, the results are shown to the creator in the form of graphical representations. Interpreting raw data can be very hard and cumbersome. Graphical representations make it convenient for the creator to analyse the performance of his videos and the feedback he has received. Various parameters are used to make comparisons such as, time of posting, Views, Like/dislike ratio, number of comments, and the ratio of positive/negative comments. These parameters are compared using graphs. Revealing some useful outcomes such as, does the time of posting the video affect the views, likes/dislikes and how toxic or useful the comments are? Do videos that have a high number of dislikes also have a higher number of negative comments? Is there a connection between the number of comments and the ratio of positive and negative comments?

Moreover, our Program generates YouTube titles from various other similar videos and suggests them to the creator. To do this, the videos of the creator will be categorized based on various parameters of the videos. One such parameter is the ratio of positive and negative comments on that video. Videos that pass a certain pre-defined threshold of this ratio are utilized. The titles of these videos are scraped. The algorithm then looks for keywords in the titles and selects them on the basis of certain parameters. It then searches for videos related to that keyword and scrapes the titles of a fixed number of videos found on the search result in order. The data collected is processed, and a new title is generated using an Artificial Intelligence model based on Transform Neural Network.

The field of social media analysis is a trending one, due to the immense data generated every second. A platform like YouTube, where 30,000 hours worth of videos are uploaded every hour, and 2 billion plus users log on every month, is a goldmine for data gathering and analysis. The users can give their reviews, and as aforementioned, our program can help rising creators understand their audience better and adapt their content accordingly. Our algorithm helps them find unique topics that are trending using various basic and advanced AI/ML algorithms, thus giving them an edge over their competitors.

Some similar models already exist that analyse user's comment on a social media platform. The purpose of the analysis can vary and can be anything from Sorting comments, understanding user behaviour, or interpreting the correlation between different parameters of a comment or video. One such model analysed millions comments on various different genres of YouTube videos and tried to find relationship between comments, views, comment ratings and topic categories. They gathered data by scraping the first 500 comments of each video, along with the user, time, and ratings or likes on the comment.

II. RELATED WORK

There have been numerous works that analyses user's comments on a Social Media platform like YouTube, Facebook, Instagram etc.

Analysing and predicting statements from comments are not new, one such model is [1]. This model has analysed over 1 million comments on 67,290 YouTube videos to find relationship between comments, views, comment ratings and topic categories. They compared the sentiment expressed by a comment to how that comment was rated by the community that is, the number of likes and dislikes for a particular comment. They predicted the ratings of comment which hadn't yet been rated by the community using the sentiment of the comments. They analysed the feasibility of utilising comments and the feedback from community to prepare models. Their work makes looking for comments better by promoting comments that are useful.

To build their dataset, they scraped the first 500 comments of each video, along with the user, time, and ratings or likes on the comment. They have gathered data like the title, tags, category, description, upload date and some other information that is made available by YouTube.

They have done a study on the distribution of comment ratings such as qualitative and quantitative work on sentiment ratings of terms in different categories. They have tried to find whether they can predict the amount of likes/dislike a comment will get based on the sentiments expressed.

There is much work done in the field of Idea generators. People have prepared some AI/ML models for the same, such as [2]. This Artificial Intelligence model utilises NLP (Natural Language Processing) to create YouTube titles or Ideas. It was trained using 20000 video titles collected from some of the largest YouTube channels. They have gathered YouTube videos of popular YouTubers by using YouTube's API. They have collected videos from different channels and used their titles as data to generate 10 new titles randomly.

This implementation uses a basic Machine Learning algorithm to generate Video titles. However, the drawback is that the generation of new titles doesn't take into account the domain of the Creator. This results in some extremely random and odd titles which are not of any relevance to the Creator. Moreover, a lot of the titles generated are grammatically incorrect and hard to interpret.

There are many works done in the field of text generation using a person's previous text pattern on social media, one of the models we came across is [3]. This model has tried to generate tweets for 10 accounts of famous people on Twitter, evaluate them using NLP (Natural Language Processing) techniques, and assess the results. They decided to look at the tweets of politicians and entertainers in particular to build a network. They used a library to mine tweets from the twitter accounts of 10 famous personalities. They collected 10,000 tweets for each public figure. They evaluated twitter data using NLP techniques like Word Frequency Analysis and Sentiment Analysis. They evaluated the style of writing a tweet of the accounts chosen. They used data to generate or predict tweets for them.

This model performs simple sentiment analysis of comments on a YouTube video [4]. The model utilizes ”Vader Lexicon”, a package present in Python. It scraps YouTube comments through YouTube's Data API. It downloads the comments and analyses them. It uses VADER (Valence Aware Dictionary for Sentiment Reasoning) to interpret Human language and emotions. The model generates a report that judges the sentiments of comments on a YouTube video. It has a dictionary of words, and each unique word has a different score. It recognizes which words are used in comments and calculates a net score. Based on the net score, it labels a particular comment in one of three categories Positive, Neutral or Negative.

While these models perform sentiment analysis on comments, they fail to draw any significant conclusion from the result. Our model will not only analyse the sentiments expressed but will also use graphs to derive some outcomes. Some relevant parameters will be compared which will generate conclusion such as the effect of factors like views, likes/dislikes on some other factors. Moreover, unlike other models, our model will rank YouTube videos based on the result of Sentiment Analysis of the comments of videos. This will allow for a more accurate ranking system of videos which have a positive response as compared to those with a negative response. As a result, videos which provide accurate information or are liked by the majority will be ranked the highest. YouTube's system of ranking videos doesn't take into account the sentiments expressed in the comment but only data such as likes/dislikes, views, and the trending topics.

Moreover, this ranking of YouTube videos based on the sentiments of their comments will be further utilised to generate new titles. This will be done by scraping relevant keywords from the titles of videos which are above a certain threshold of positive comments, and search for other YouTube videos related to that keyword. The titles of these YouTube videos will be used to suggest new Titles and Ideas to the Creator. This is a more accurate way of suggesting new Titles as this filters the videos which have a negative response overall and suggest Trending titles which have a high chance of going viral.

III. PROPOSED METHODOLOGY

Our goal is to perform sentiment analysis on the comments of YouTube videos using a logistic regression model and filtering out videos with a positive response. Then, we aim to extract keywords, fetch videos related to those keywords and use this to form our dataset. By using a neural network on this data set, we want to generate or predict new video titles. Fig. 1, shows the process in detail.

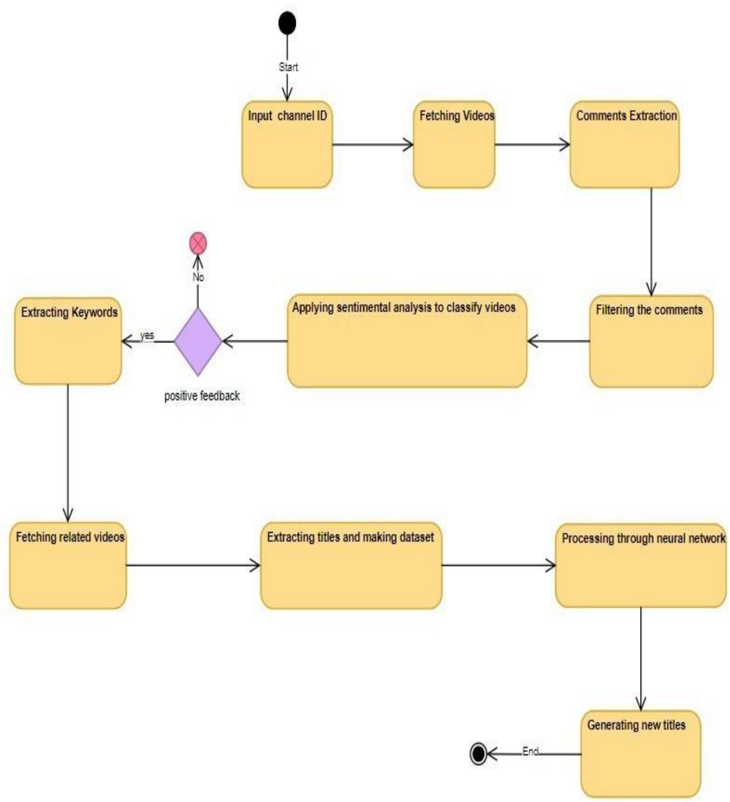

Fig. 1. A simplified flow chart of the Process to generate new you tube titles based on sentimental analysis

Sentiment Analysis is one of the Natural Language Processing techniques, which can be used to determine the sentiments of text. This helps us filter out comments and find which ones are useful. We have used the YouTube API to extract comments.

For extraction and pre-processing of the first part of our project, we have imported various modules like us, re, json, demoji, pandas and nltk. By providing the api key and channel id, we were able to access multiple videos and extract multiple comments. We have utilized the demoji function to find emojis and replace them with a blank space. Next, we have detected the language of comments using the 'detect' function from the module langdetect so we can extract the comments that are in the English language only. We have done this to maintain consistency in our dataset. We have also removed special characters like %, &, $ etc. Moreover, we removed stop words from it, stripped trailing spaces and converted all comments to lowercase.

Next, we found the polarity of the comments for our test data. Polarity is what tells us whether a comment is positive or negative. Using the TextBlob function, we found the polarity score of each comment. We set a threshold to classify the comments based on their polarity. We assigned sentiment of each comment with polarity > 0 as 1 or positive and that for polarity <= 0 as -1 or negative. We then converted this data set to .csv format.

For pre processing of the second part, we started by extracting video titles of YouTube videos that showed up in search results of keywords. We filtered our data in the same way as for part 1. We removed emojis, extracted English titles, removed special characters, converted to lower case, removed trailing spaces, and removed 'stop words. We then converted this data set to .csv format.

To perform sentiment analysis, we used a logistic regression model. Logistic Regression is an algorithm that is used for Classification of data. It used for classifying or grouping data based on a criteria. In our model, the criteria is whether a comment under a YouTube video is positive or negative. Our task was to train a model such that it can give a binary output which helps identify the class of new data.

To generate video titles, we could have taken a few different approaches including gpt-2, lstm, BERT and Elmo.

Lstm or Long Short-Term Memory is a form of Artificial Neural Network in which RNNs learn from not only the current state, but the output of the previous [5] has followed this approach. This is one path we could have followed. To generate text using lstm, a model learns the Probability of a what the next word is going to be. It does this based on the previous series of words used in the text. Language models can exist at a few different levels. They include character, anagrams, full sentences, or large paragraphs. The model takes input data and tries to predict the next word based on the patterns in the text given in the input. In Recurrent Neural Networks like this, the text from previous outputs is used as the input for the next word. Therefore, a loop is created in the neural network. This enables neurons to take into consideration all the knowledge learned so far.

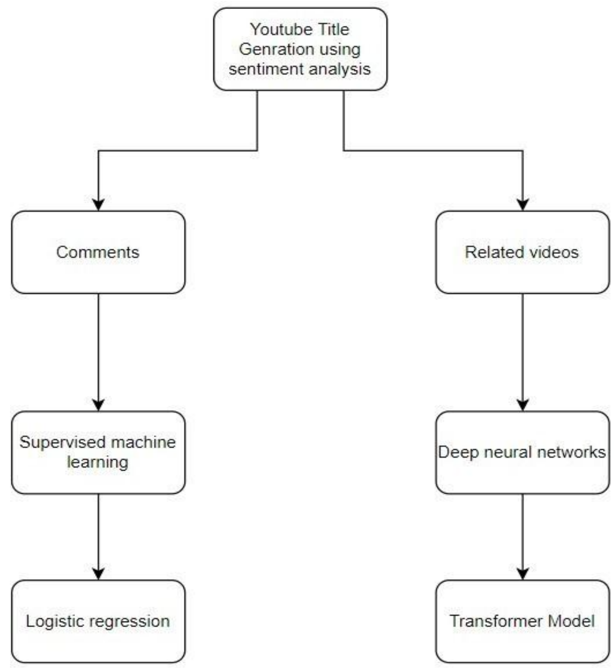

Fig. 2. Major AI/ML components of the programme

However, the model we ended by choosing was gpt-2. We started by importing gpt-2. gpt-2 is an open-source artificial intelligence that can translate and summarise text, answer questions, most importantly, and generates text output which is identical to humans. GPT-2 has over 1.5 billion parameters and was trained on a dataset of 8 million web pages or about 40GB of text. gpt-2 is able to predict the next word or sentences, after taking into account all previous words given within some text. The diversity of the dataset causes this simple goal to contain naturally occurring demonstrations of many tasks across diverse domains.

The method used by gpt-2 to generate text varies from other similar packages. We chose gpt-2 because it has the edge when it comes to certain important criteria. As a whole, gpt-2 can sustain context over the entire text generation. This makes it good for generating titles that are not only relevant but also logically sensible. The text generated is also correct grammatically for the most part, and barely has any typos. The fact that gpt-2 was trained using a lot of sources means the model can also include unconventional phrases in the input text. Though gpt-2 can only generate a maximum of 1024 tokens per request, this drawback is irrelevant as video titles are extremely short. Higher temperatures allow for more interesting texts which is why we have kept our temperature in the range of 0.7-0.9. Due to GPT-2's architecture, it performs well with more powerful GPUs and scales up with the power of GPUs.

We used the 124M model in gpt-2 to perform this. We read the csv files containing our data set of titles using pandas.

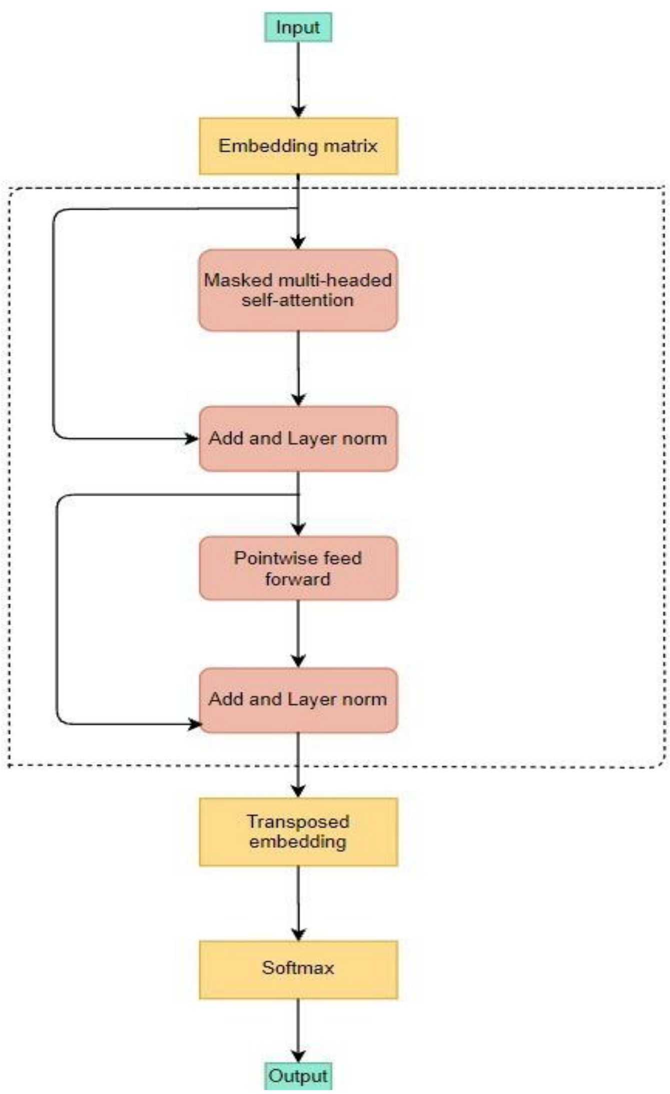

Fig. 3. GPT-2 Tranform Neural Network Model

IV. RESULT

We used Google Colabs and Jupyter Notebook for all the training. We started sentiment analysis with over 1,200 total comments under different YouTube videos. We started with pre-processing the data and then proceeded to train it. We classified the polarity of the comments as -1 or 1. After finding the polarity, we split our data into 2 parts – test data and training data. 20% of our data was taken as test data and the rest 80% was taken as training data. Then, using the training data we train our model, and use it over the test data to predict the polarities.

We decided to use f1 score as our benchmark over other determinents like recall, precision, and accuracy. Precision is the ratio of correctly predicted positive cases out of all the predicted courses. It is a relevant metric when it is important to minimize the amount of false positives which is true for our model. Recall is the ratio of correctly predicted positive cased out of all the positive cases. It is relevant when it is important to minimize the number of false negatives. It is also certainly important for our model but arguably not as important as precision. accuracy is the ratio of all the correctly predicted cases. It is relevant for our model as all classes hold importance. F1 score is the harmonic mean of Precision and Recall. Its advantage is that it provides a better measurement of the cases that were incorrectly classified as compared to the Accuracy. This just means that the f1 score gives more importance to the number of false positives and false negatives. Another advantage to the f1 score is that it is a far more consistent metric when the class distribution is disproportional. In fact, In almost all linear regression models including ours, class distributions are not balanced. Due to this, F1-score is an important metric to evaluate our model.

We had 1244 true positives, 32 false negatives, 110 false positives and 68 true negatives. Our confusion matrix after this step is shown in Fig. 4.

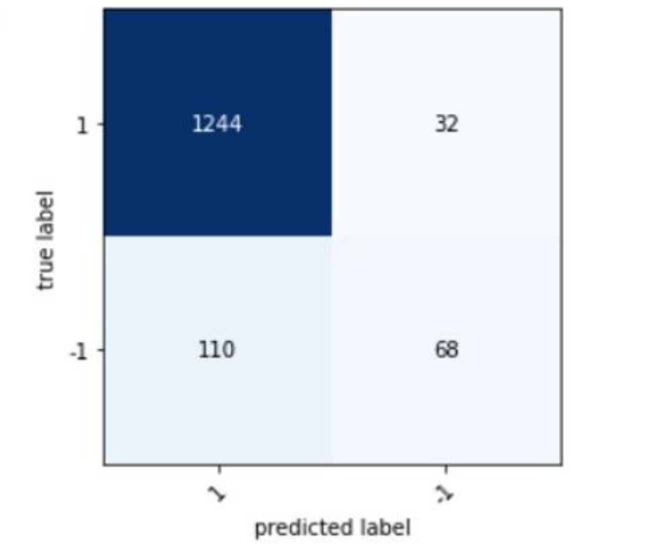

Fig. 4. First Confusion Matrixl

Scores for our run are shown in Fig. 5 and Fig. 6.

	precision	recall	f1-score	support
-1	0.68	0.38	0.49	178
1	0.92	0.97	0.95	1276
accuracy			0.90	1454
macro avg	0.80	0.68	0.72	1454
weighted avg	0.89	0.90	0.89	1454

Fig. 5. First Score Table

0.7176081188281314

Fig. 6. First F1 Score

We improved on this iteration by removing all the words that are not in English language. We did this using the 'detect' function from the module langdetect to detect and then extract the comments that are in the English language only. Our f1 score increased to 0.87. We had 756 true positives, 120 false negatives, 60 false positives and 526 true negatives. Our confusion matrix and f1 score are as shown in Fig. 7.

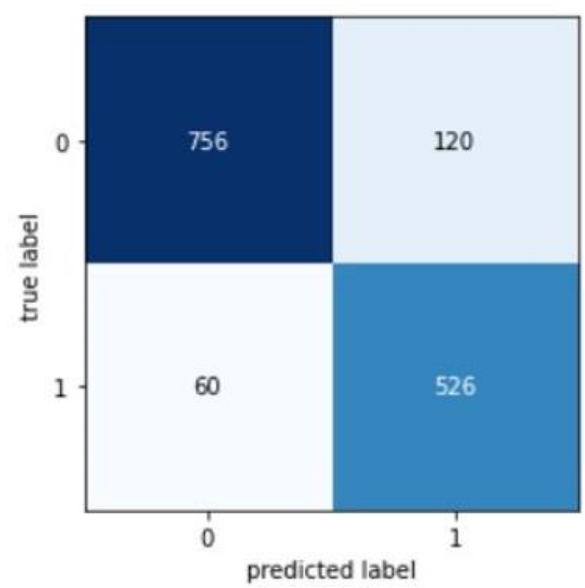

Fig. 7. Second Confusion Matrix

Scores for our run are shown in Fig. 8 and Fig. 9.

	precision	recall	f1-score	support
-1	0.81	0.90	0.85	586
1	0.93	0.86	0.89	876
accuracy			0.88	1462
macro avg	0.87	0.88	0.87	1462
weighted avg	0.88	0.88	0.88	1462

Fig. 8. Second Score Table

0.8737565625863499

Fig. 9. Second F1 Score

To improve our f1 score further, we further proceeded by converting the same dataset to all lowercase letters, and removing all stop words i.e., unnecessary words from it.

The plot in Fig. 10, shows the distribution of positive (1) and negative (-1) comments in our analysis of the dataset.

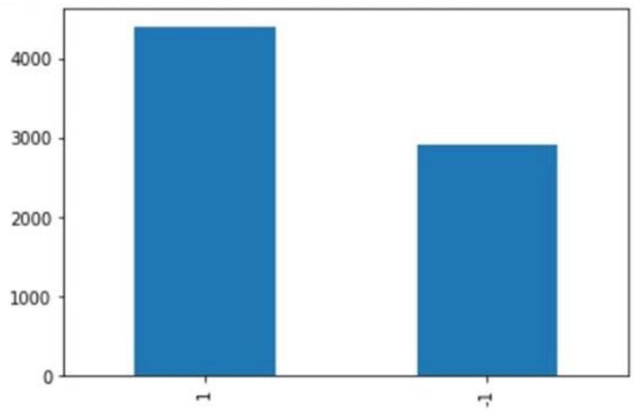

Fig. 10. Frequency Bar Plot

We had 769 true positives, 94 false negatives, 38 false positives and 561 true negatives. Our confusion matrix is as shown in Fig. 11.

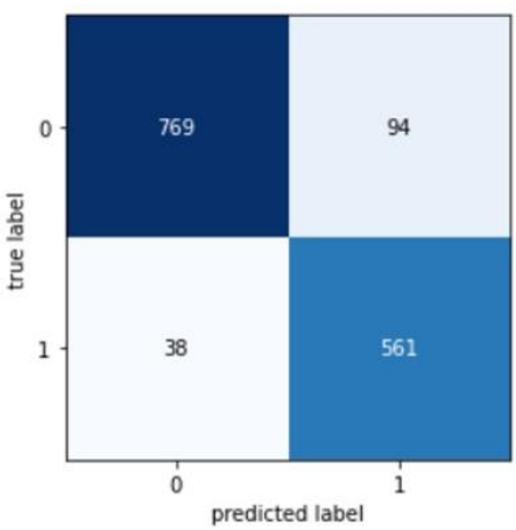

Fig. 11. Third Confusion Matrix

Scores for our run are shown in Fig. 12 and Fig. 13.

	precision	recall	f1-score	support
-1	0.86	0.94	0.89	599
1	0.95	0.89	0.92	863
accuracy			0.91	1462
macro avg	0.90	0.91	0.91	1462
weighted avg	0.91	0.91	0.91	1462

Fig. 12. Third Score Table

```
0.9078474629687994
```

Fig. 13. Third F1 Score

For title prediction, our results for the first iteration are as Fig. 14.

```
brine shrimp venom baby red herobrine ward
====================
brine iron golem minecraft golem next
====================
```

Fig. 14. First Prediction

Some of the titles generated here do not make sense, so we tried to improve our model. We increased our temperature to 0.9 to increase the randomness of words that are utilized in our titles. We also increased the size of our dataset for a single keyword. The results are shown in Fig. 15 and Fig. 16.

```
brine venom poppy playtime minecraft shorts found rarest iron golem
====================
```

Fig. 15. Second Prediction 01

```
craft one super long zombie trying diwali firecrackers minecraft monster
====================
```

Fig. 16. Second Prediction 02

Clearly, there is more randomness and a greater number of words that are being used.

In our last iteration, we tried using multiple different keywords simultaneously to obtain titles. The results are shown in Fig. 17 and Fig. 18.

```
[198 | 1056.43] loss=0.01 avg=0.13
[199 | 1060.81] loss=0.02 avg=0.13
[200 | 1065.20] loss=0.01 avg=0.13
Saving checkpoint/run1/model-200
```

Fig. 17. Loss and Average Loss

```
Call of Duty League 2021 Season Stage IV Major Tournament Day 4TIMTHETATMAN PLAYS
====================
This is the worst selling Xbox Games of 2011 in Xbox Live Gold Members...This is the worst selling
====================
```

Fig. 18. Third Prediction

V.CONCLUSION

Our model proves that implementing sentiment analysis and title generation on a Social Media site like YouTube is possible and useful. Our model scraped comments from the comment section of various YouTube videos and performed Sentiment Analysis. We performed pre-processing on our data set by using modules such as os, re, json, demoji, pandas and nltk. We were able to successfully find the polarity of comments after pre processing by removing various junk. We also classified the comments based on their polarity score by setting a threshold score of 0. We tested model by creating the confusion matrix and calculating f1 score. Our f1 score increased from 0.72 to 0.91 as made adjustments and filtered our data set.

Moreover, we were also able to scrape video titles and use trending keywords to predict video titles of new videos. We settled on a gpt-2 model due to its various advantages such as being able to produce sustained text with context over the entire length of text generation.

REFERENCES

[1] S. Siersdorfer, S. Chelaru, W. Nejdl, and J. San Pedro, "How useful are your comments? analysing and predicting youtube comments and comment ratings," *https://www.researchgate.net/publication/221023733*, 2010.

[2] gdemos0, "Youtube video idea generator ai." *https://github.com/gdemos01/YoutubeVideoIdeasGeneratorAI/*, 2010.

[3] S. T. Jarlai Morris, Nataliia Nevinchana, "How well can our re- current neural networks approximate tweets of our political figures and entertainers?" *https://towardsdatascience.com/tweet-generation-with- neural-networks-lstm-and-gpt-2-e163bfd3fbd8/*, 2020.

[4] Anshnarad, "Youtube comment analysis,"*https://github.com/Anshnarad/Youtube-Comment-Analysis/*, 2019.

[5] somang1418,"Youtube-title-generator,"*https://github.com/somang1418/Youtube-Title-Generator*, 2020.