Commentator - YouTube Comment Analysis using Natural Language Processing

Sanjay Kumar R, Kumaresh P,
Prabhu M
Department of Computer Science and Engineering
Panimalar Engineering College
Chennai, India
sanjaykumar1990112@gmail.com,
kumareshp2001@gmail.com,
prabhumurugavel2000@gmail.com

Abstract- With the rise of social media and public data, opinion mining is more accessible than ever. It is valuable for content creators, businesses, and advertisers to better understand what users are thinking and feeling. Commentators (Analysis of YouTube comments using natural language processing) examine comments on YouTube videos and create a Naive Bayes classifier to automatically determine how they feel. The naive Bayesian taxonomist classifies feelings into six types of human emotions: Sadness, Anger, Surprise, Fear, Happiness, and Love. Categorizing comments into these categories represents people's opinions more accurately than categorizing them as positive or negative comments. Based on commenters' analysis of the video, viewers can also understand the nature of the video and can also choose not to watch that particular video if they're not interested. Commenters can also uncover advertising made by content creators or inaccurate facts mentioned by content creators through analysis of YouTube video comments.

Keywords—Multinomial Naïve Bayes Model, datasets

I. INTRODUCTION

Emotions and opinions play an important role in human behaviour as we often make decisions based on the experiences and thoughts of others. Since people's opinions are subjective and influenced by their personal experiences and beliefs, learning about them helps to gain a broader perspective. Public opinion has long been interested in organizations, businesses and politicians. The collective opinion of the people can help predict who will win an election, generating knowledge about future trends and topics. It also helps to define a common view of products and services, helps the marketing team decide on marketing strategies, improves existing products or new production, and improves customer support. Therefore, it is essential to perceive sensations in different areas.

Initially, citizens' opinions are collected through surveys or questionnaires and reviewed manually. However, as Internet usage increased, people started voicing their opinions online more often. The explosion of social networking platforms in recent years has allowed users to share all kinds of information and introduced more ways and ways of expressing their opinions. Blogs, discussion forums, reviews, comments, and microblogging services, such as Twitter or Facebook, are rich sources of data, including audio files, videos, images, and opinions.

One of these platforms, YouTube, is currently one of the most popular social media platforms in the world. It allows users to share their videos or watch videos uploaded by other users and provide feedback. Users can provide feedback by favouring or rejecting the video by rating it positive or negative, respectively. Users can also provide written feedback in the form of posted comments on the video. Each video can contain thousands of comments, making YouTube a platform.

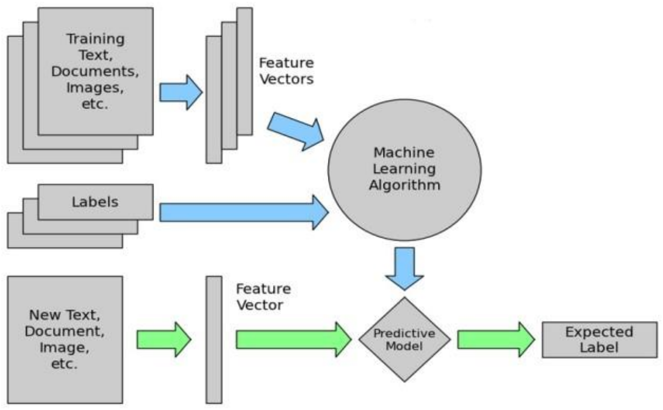
II. EXISTING SYSTEM

The major flaw in the existing system (YouTube) is that the nature of the comments of a particular YouTube video is not displayed and to add fire to flame the dislike count of a particular YouTube video is removed by Google to stop spreading hatred and it is visible only to the content creators. The removal of dislike button means that the viewers need to go through the top comments of the YouTube video to get the gist of the nature of the video.

III. PROPOSED SYSTEM

The proposed system is a Multinomial Naive Bayes Model with a classifier that classifies the comments of a YouTube video into 6 categories: Sad, Angry, Surprise, Fear, Happy and Love. Commentator thus reduces the hassle of viewers to go through each and every comments of the comment section to determine the nature of the video. Commentator analyses the nature of the video by extracting the comments with the help of a YouTube API which returns the different parameters of YouTube Video as a response from which comments are extracted. The Multinomial Naive Bayes classifier is trained with the training dataset of 9,000 values for the prediction of the accurate sentiment expressed in the comments.

IV. SYSTEM ARCHITECTURE OVERVIEW


Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
Labels
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Expected Label

V. LIBRARIES/API's USED

Pandas- Pandas is an open source library built on top of the NumPy library. It is a Python package that provides various data structures and operations for dealing with numeric and time series data. It is mainly popular for much easier data entry and analysis. Pandas is fast and gives high performance and productivity to the users.

Pandas Series- A Pandas string is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively known as the index. The Pandas series is nothing but a column in an Excel sheet. Tags need not be unique, but must be hashable. The object supports integer-based and label-based indexing and provides a wide range of methods to perform index-related operations.

Pandas Dataframe- Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the data, rows, and columns.

Natural Language Toolkit (NLTK)- The NLTK module is a huge toolkit designed to help you with the entire Natural Language Processing (NLP) approach. NLTK will give you everything from splitting paragraphs into sentences, splitting words, identifying parts of speech, highlighting topics, and even helping your machine understand what the text is about.

Corpus — The body of the text, singular. Corpora is its plural. Example: A collection of medical journals.

Lexicon — vocabulary and its meaning. For example: English dictionary. However, considering that there are different thesaurus in each field. For example, for financial investors, the first meaning of the word Bull is the person who is confident in the market. Compared with the "common English vocabulary", the first meaning of the word is animal. Therefore, financial investors, doctors, children, mechanics, etc. all have a special vocabulary.

Token — Each "entity" is part of a rule based split. For example, when a sentence is "split" into words, each word is a tag. If you split a paragraph into sentences, each sentence can also be a marker.

NLTK Modules- The Natural Language Toolkit (NLTK) is a set of libraries and programs in Python for natural language processing tasks. It is one of the most used Python NLP libraries. It can perform various NLP tasks such as encryption, root generation, POS tagging, lemmatization, and classification, among others.

Language Processing Task	NLTK module
Accessing Corpora	nltk.corpus
String Processing	nltk.tokenize, nltk.stem
Collocation discovery	nltk.collocations
Part-of-speech tagging	nltk.tag
Classification	nltk.classify, nltk.cluster
Chunking	nltk.chunk
Parsing	nltk.parse
Semantic interpretation	nltk.sem, nltk.inference
Evaluation metrics	nltk.metrics
Probability and Estimation	nltk.probability
Applications	nltk.app, nltk.chat
Linguistic fieldwork	nltk.toolbox

Scikit-learn- (Sklearn) is the maximum beneficial and strong library for gadget gaining knowledge of in Python. It offers a choice of green equipment for gadget gaining knowledge of and statistical modeling along with classification, regression, clustering and dimensionality discount through a consistence interface in Python. Sci-package study functions diverse classification, regression, and clustering algorithms along with aid vector machines, random forests, gradient boosting, k-means, and is designed to interoperate with the Python numerical and medical libraries NumPy and SciPy. It is certified below a permissive simplified BSD license and is shipped below many Linux distributions, encouraging educational and industrial use.The library is constructed upon the SciPy (Scientific Python) that should be mounted earlier than you may use scikit-study. This stack includes:

NumPy: Base n-dimensional array package

SciPy: Fundamental library for scientific computing

Matplotlib: Comprehensive 2D/3D plotting

IPython: Enhanced interactive console

Sympy: Symbolic mathematics

Pandas: Data structures and analysis

YouTube API- The YouTube Application Programming Interface (YouTube API) lets in builders to get entry to video information and YouTube channels statistics through kinds of calls, REST and XML-RPC. Google describe YouTube`s API assets as "APIs and Tools that permit you to carry the YouTube revel in on your webpage, utility or device.

YouTube API Requirements- You want a Google Account to get entry to the Google API Console, request an API key, and sign up your utility.Create a mission withinside the Google Developers Console and acquire authorization credentials so your utility can post API requests.After growing your mission, ensure the YouTube Data API is one of the offerings that your utility is registered to use:Go to the API Console and pick out the mission which you simply registered.Visit the Enabled APIs page. In the listing of APIs, ensure the popularity is ON for the YouTube Data API v3.

If your utility will use any API techniques that require consumer authorization, study the authentication manual to discover ways to put in force OAuth 2.zero authorization. Select a customer library to simplify your API implementation.Familiarize your self with the center ideas of the JSON (JavaScript Object Notation) statistics layout. JSON is a common, language-impartial statistics layout that gives a easy textual content illustration of arbitrary statistics structures.

YouTube API CommentThreads: list- CommentThreads: list is a part of YouTube API and returns a list of comment threads that match the API request parameters.The table below shows common use cases for this method. You can click on a use case name to load sample parameter values in the APIs Explorer.

Use cases

list (by video ID)	This example retrieves all comment threads associated with a particular video. The request's videoId parameter identifies the video.
list (by channel ID)	This example retrieves all comment threads about the specified channel. The request's channelId parameter identifies the channel. The response does not include comments left on videos that the channel uploaded.
list (all threads related to channel ID)	This example retrieves all comment threads associated with a particular channel. The response could include comments about the channel or about the channel's videos. The request's allThreadsRelatedToChannelId parameter identifies the channel.

Filters of YouTube API-

allThreadsRelatedToChannelId: string

The allThreadsRelatedToChannelId parameter instructs the API to go back all remark threads related to the desired channel. The reaction can encompass feedback approximately the channel or approximately the channel`s movies.

channelId: string

The channelId parameter instructs the API to go back remark threads containing feedback approximately the desired channel. (The reaction will now no longer encompass feedback left on movies that the channel uploaded.)

id: string

The channelId parameter instructs the API to go back remark threads containing feedback approximately the desired channel. (The reaction will now no longer encompass feedback left on movies that the channel uploaded.)

videoId: string

The videoId parameter instructs the API to return the comment strings associated with the specified video ID.

maxResults: unsigned integer

The maxResults parameter specifies the maximum number of items to be returned in the result set. This parameter is not supported for use with the id parameter. Accepted values are from 1 to 100, inclusive. The default value is 20.

moderationStatus: string

This parameter can only be used in a properly authorized request. Set this parameter to limit comment streams that are returned to a specific moderation state. This parameter is not supported for use with the id parameter. Default is published. Acceptable values are:

heldForReview – Retrieve comment threads awaiting review by moderators. A comment thread can be included in an answer if a top-level comment or at least one of the replies to that comment is pending review.

likelySpam – Retrieve comment threads classified as likely spam. A comment thread may be included in an answer if the top-level comment or at least one of the replies to that comment is considered potentially spam.

published – Retrieve published comment threads. This is the default. A comment thread can be included in an answer if its top-level comment has already been posted.

order: string
The order parameter specifies the order in which the API response should list the commented strings. This parameter is not supported for use with the id parameter. Valid values are:

time - Comment threads are ordered by time. This is the default behavior.

relevance - Comment threads are ordered by relevance.

pageToken: string

The pageToken parameter identifies a specific page in the result set that should be returned. In an API response, the nextPageToken property identifies the next page of the result that can be retrieved. This parameter is not supported for use in conjunction with the id parameter. The searchTerms parameter instructs the API to limit the API response to only contain comments that contain the specified search terms. This parameter is not supported for use in conjunction with the id parameter.

searchTerms: string

The searchTerms parameter instructs the API to limit the API response to only comments containing the specified search terms. This parameter is not supported for use with the id parameter.

textFormat: string

Set the value of this parameter to html or plainText to instruct the API to return comments left by the user in HTML or plain text format. Default is html.
Acceptable values are:

html – Returns the comments in HTML format. This is the default value.

plainText – Returns the comments in plain text format.

YouTube API Response Structure- If successful, method returns a response body with the following structure:

```
{
  "kind": "youtube#commentThreadListResponse",
  "etag": etag,
  "nextPageToken": string,
  "pageInfo": {
    "totalResults": integer,
    "resultsPerPage": integer
  },
  "items": [
    commentThread Resource
  ]
}
```

VI. MODULE DESIGN SPECIFICATION

Dataset Importing Module - The dataset importing module is the first step where the training dataset and testing dataset for the Multinomial Naive Bayes Model are uploaded in the Google Colaboratory and these files need to uploaded everytime the runtime is restarted or in other words the particular window is closed. The imported data is classified as Comment and Emotion with the help of Pandas DataFrame.

NLTK Libraries Importing Module- The NLTK Libraries Importing Module is where all the NLTK libraries required for data preprocessing of both the training and testing dataset are imported. The Libraries imported are NLTK RegexpTokenizer to tokenize the words in sentences of the training and testing datasets, NLTK PorterStemmer to stem the tokens obtained from the sentences of the training and testing datasets and NLTK Stopwords Remover to remove the common stopwords of the English Language from the tokenized stemmed words of the training and testing datasets.

Dataset Cleaning Module- The Dataset Cleaning Module is where the imported NLTK Libraries such as NLTK RegexpTokenizer, NLTK PorterStemmer and NLTK Stopwords Remover are used to clean both the training and testing dataset. The datasets are first converted to lower case and converted to tokens by the use of lower() and tokenize() functions. The stopwords are removed followed by the stemming of tokens in training and testing dataset. Finally, the clean training and testing datasets are sent to the next module for Transformation of Data.

Dataset Transformation Module- The Dataset Transformation Module is where the training and testing datasets are transformed into vectors to be trained and tested by the Multinomial Naive Bayes Model. The CountVectorizer and TfidfTransformer are imported from the scikit-learn(sklearn) library for which the objects are first created. The clean training and testing datasets are converted into vectors by the means of CountVectorizer and the terms are given importance or ignored by the means of TfidfTransformer. The fit_transform() method of both CountVectorizer and TfidfTransformer is used for the transformation of training data and transform() method of both CountVectorizer and TfidfTransformer are used for transformation of testing data.

Model Building And Training Module- The Model Building Module is where the Multinomial Naive Bayes Model is built by importing MultinomialNB from the scikit-learn(sklearn) library. The object is created for MultinomialNB that is the model is successfully built. The fit() method is used to train the model with the cleaned and vectorized training dataset and labelled dataset and system resources are utilized by the model to be trained such as RAM and Disk Space and after the model is trained it will able to classify the nature of the YouTube comments into one of these categories: Sad, Happy, Angry, Fear, Love and Surprise.

YouTube Comment Extraction Module- The YouTube Comment Extraction Module is where the comments of a particular YouTube video entered by the users are extracted for prediction of sentiment by the trained Multinomial Naive Bayes Model. The request to the YouTube API is created with the means of API Key and Video ID and Video ID is accepted as input from the user. From the YouTube API response, comments are extracted and stored in a list. The list is then cleaned and transformed into vectors for the trained Multinomial Naive Bayes Model to predict the sentiment.

Sentiment Prediction Module- The Sentiment Prediction Module is where the sentiments of comments extracted by the YouTube API and classified and the sentiments of the testing dataset are predicted by the trained Multinomial Naive Bayes Model. The cleaned and transformed testing dataset and extracted YouTube video comments are predicted by the model by the means of predict() method to which the datasets are passed as arguments.

Performance Analysis Module- The Performance Analysis Module is where the model is assessed based on the various performance parameters. The performance parameters include Accuracy Score, Classification Report and Confusion Matrix. The various libraries required for accessing the performance of the trained Naive Bayes Model are imported from the scikit-learn(sklearn) library. The objects are created for the performance paramteters and finally the various performance parameters of the trained Multinomial Naive Bayes Model are displayed.

VII. NAÏVE BAYES ALGORITHM

Bayes' Theorem- Naive Bayes is based on Bayes' theorem, where the Naïve adjective states that the properties of a data set are mutually independent. The occurrence of one feature does not affect the probability of the occurrence of the other feature. For small sample sizes, Naïve Bayes may outperform more robust alternatives. Relatively powerful, easy to implement, fast and accurate, it is used in many different fields. For example, email spam filtering, disease diagnosis, treatment decision making, RNA sequencing in taxonomic studies, to name a few. However, we have to keep in mind about the type of data and the type of problem to be solved that dictates which classification model we want to choose. Strong violations of the independence assumptions and nonclassification problems can lead to poor performance. In practice, it is recommended to use different classification models on the same dataset and then consider the performance, as well as computational efficiency. To understand how Naïve Bayes works, first, we have to understand the concept of Bayes` rule. This probability model was formulated by Thomas Bayes (17011761) and can be written:

$$Posterior\ Probability = \frac{Conditional\ Probability * Prior\ Probability}{Predictor\ Prior\ Probability}$$

$$P\left(\frac{A}{B}\right) = \left(\frac{P(A \cap B)}{P(B)}\right) = \frac{P(A) * P\left(\frac{B}{A}\right)}{P(B)}$$

where,
PA= the prior probability of occurring A
PBA= the condition probability of B given that A occurs
PAB= the condition probability of A given that B occurs
PB= the probability of occuring B

The posterior probability, can be interpreted as: What is the revised probability of an event occurring after taking new information into consideration? It is a better reflection of the underlying truth of a data generating process because it includes more information.

Conditional Probability- The probability of one event A occurring when another event B with some relationship to A has already occurred is called conditional probability.

$$P\left(\frac{B}{A}\right) = \left(\frac{P(A \cap B)}{P(A)}\right)$$

This expression is valid only when P(A) is greater than zero. Multinomial Naïve Bayes uses term frequency i.e. the number of times a given term appears in a document. Term frequency is often normalized by dividing the raw term frequency by the document length. After normalization, the term frequency can be used to calculate the maximum likelihood estimates based on the training data to estimate the conditional probability.

Prior Probability- This probability can be defined as prior knowledge or belief, i.e. the probability of an event is calculated before new data is collected. This probability is modified as new information becomes available to produce more accurate results. If prior observations are used to calculate probability, we call it pre-probability.

VIII. PERFORMANCE ANALYSIS

True Positives/Negatives-

	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

True positives and negatives are actually correctly predicted observations and are therefore shown in green. We want to minimize false positives and false negatives so that they are shown in red. These terms are a bit confusing. So let's take each term one by one and fully understand it.

The True Positive value (TP) is the correctly predicted positive value, which means that the actual class value is yes and the predicted class value is also yes. Eg. if the actual class value indicates that this passenger survived and the predicted class tells you so.

True negatives (TN) are correctly predicted negative values, which means that the actual class value is nil and the predicted class value is also nil. Eg. if the actual class indicates that this passenger did not survive and the predicted class tells you so.

False positives and false negatives, these occur when your actual class conflicts with the predicted class.

A false positive (FP) result is when the actual class is no and the predicted class is yes. Eg. if the actual class tells you that this passenger did not survive, but the predicted class tells you that this passenger will survive.

When the actual class is yes but the predicted class is not. Eg. if the actual class value tells you that this passenger survives and the predicted class tells you that the passenger will die.

False Negatives (FN is when the actual class is yes but the predicted class is not. Eg. if the actual class value tells you that this passenger survives and the predicted class tells you that the passenger will die. Once you understand these four parameters, we can calculate accuracy, precision, recall and F1 score.

Accuracy- Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, you have to look at other parameters to evaluate the performance of your model.

Accuracy = TP+TN/TP+FP+FN+TN

Precision- Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate.

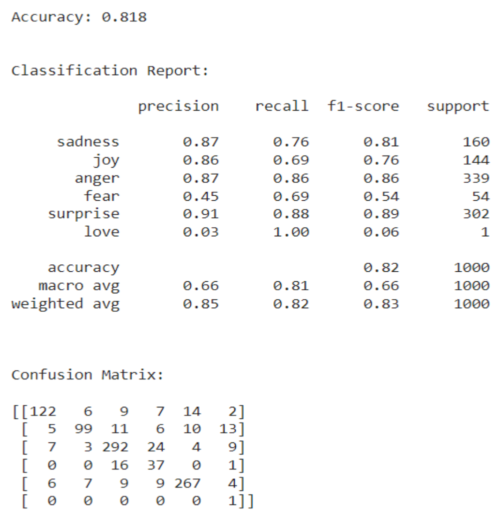
Precision = TP/TP+FP

Recall (Sensitivity)- Recall is the ratio of correctly predicted positive observations to the all observations in actual class yes. The question recall answers is: Of all the passengers that truly survived, how many did we label?

Recall = TP/TP+FN

F1 Score- The F1 score is the weighted average of precision and recall. Therefore, this score takes into account both false positives and false negatives. Visually, it's not as straightforward as precision, but F1 is often more useful than precision, especially if you have unequal class distributions. Accuracy works best if false positives and false negatives have the same cost. If the costs of false positives and false negatives are very different, it is best to consider both accuracy and recall.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)



```
Accuracy: 0.818

Classification Report:

              precision    recall  f1-score   support

     sadness       0.87      0.76      0.81       160
         joy       0.86      0.69      0.76       144
       anger       0.87      0.86      0.86       339
        fear       0.45      0.69      0.54        54
    surprise       0.91      0.88      0.89       302
        love       0.03      1.00      0.06         1

    accuracy                           0.82      1000
   macro avg       0.66      0.81      0.66      1000
weighted avg       0.85      0.82      0.83      1000


Confusion Matrix:

[[122   6   9   7  14   2]
 [  5  99  11   6  10  13]
 [  7   3 292  24   4   9]
 [  0   0  16  37   0   1]
 [  6   7   9   9 267   4]
 [  0   0   0   0   0   1]]
```

IX. CONCLUSION

In this project, the aim was to construct a YouTube comment dataset and build a Multinomial Naive Bayes Model to classify YouTube comments as one of the following categories: Sad, Happy, Angry, Fear, Love, Surprise. A total of 11,000 comments were manually labelled based on their sentiment which are equally divided based on one of the above categories. Following that, the performance of Multinomial Naive Bayes Model on the YouTube comment dataset was evaluated and the accuracy was found to be 81.8%. It was also observed that the sentiments of comments expressed by the people in the YouTube Comments section is more expressive when categorized into more categories rather than positive and negative. With the help of the analysis of Comments of YouTube video from Commentator, viewers can clearly understand the true nature of the opinions expressed by the people who viewed the video.

YouTube content creators and other media agencies can also use this system to analyze people's opinions expressed for a video. This system also serves as an alternative solution to the removal of dislike button as of 2022. Thus, different categories of videos are tested and the sentiments of the videos and result was that positive videos are found to be dominant in emotions Happy, Surprise, Love and negative videos are found to be dominant in emotions Anger, Fear and Sad.

X. FUTURE ENHANCEMENTS

The future enhancements that can be done to the Multinomial Naive Bayes Model to furthermore analyse and improve the prediction of sentiments expressed by people in Comment Section are:

Increasing the number of training and testing datasets and implementing more classification labels in training and testing datasets to increase the overall accuracy of the sentiment predicted by the model and to to give a much more brief view of the emotions expressed by the viewers respectively.

Creating a more complex dataset rich in vocabulary and long sentences to further increase the accuracy of the sentiment predicted by the model

Implementing the model in a user friendly GUI interface though an Android App or Website for better user experience and for non-power users to be able to use the model effectively.

X. REFERENCES

[1] C. Day, "The importance of sentiment analysis in social media analysis." [Online]. Available: https://www.linkedin.com/pulse/ importance-sentiment-analysis-social-media-christine-day

[2] B. Liu, Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers, 2012.

[3] C. C. Aggarwal and C. Zhai, A Survey of Text Classification Algorithms. Boston, MA: Springer US, 2012, pp. 163–222.

[4] B. Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, 2nd ed. Cambridge University Press, 2020.

[5] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, “Text classification algorithms: A survey,” Information, vol. 10, no. 4, 2019.

[6] “Youtube partner program overview & eligibility,” https://support.google. com/youtube/answer/72851?hl=en, accessed: 2021-02-09.

[7] “How to earn money on youtube,” https://support.google.com/adsense/ answer/72857?hl=en, accessed: 2021-01-05.

[8] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[9] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? sentiment classification using machine learning techniques,” in Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002). Association for Computational Linguistics, Jul. 2002, pp. 79–86.

[10] M. S. Neethu and R. Rajasree, “Sentiment analysis in twitter using machine learning techniques,” in 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp. 1–5.