

YOUTUBE DATA AND SENTIMENT ANALYZER

PROJECT REPORT

Submitted By:

Sahiba Bedi

16BCE0768

Rishab Gupta

16BCE0757

Course Title: Social & Information Networks

Course Code: CSE3021

Guided By:

Prof. Rishin Haldar

School of Computer Science and Engineering



October 2018

CERTIFICATE

This is to certify that the project work entitled **“Youtube Data and Sentiment Analyzer”** that is being submitted by **Sahiba Bedi** and **Rishab Gupta** for the course **CSE3021 - “Social Information Networks”** is a record of bonafide work done under my supervision. The contents of this project work in full or in parts, have neither been taken from any other source nor have been submitted for any other course.

Signature of Faculty:

Prof. Rishin Haldar

ACKNOWLEDGEMENT

We'd like to show our gratitude towards **Prof. Rishin Haldar** for his immense help throughout the project and always being there for us. We would also like to thank the **Dean of SCOPE** and the University Management for giving us the opportunity to carry out our study at the University.

Submitted By:

Sahiba Bedi

Rishab Gupta

ABSTRACT

For the past several years YouTube has been by far the largest user-driven online video provider. While many of these videos contain a significant number of user comments, little work has been done to date in extracting trends from these comments because of their low information consistency and quality. In this paper we perform sentiment analysis of the YouTube comments related to popular topics using natural language processing techniques as well as scraping data of popular YouTube videos based on their video IDs as well as scraping data of top 10 videos based on the search term input by the user. We demonstrate sentiment using a mathematical function and training our classifier which will differentiate the comment tokens into their degree of positiveness and negativeness. This also takes into account the emoji's used by the users while commenting on videos.

INTRODUCTION

With the rapid growth of social networking and the internet in general, YouTube has become by far the most widely used video-sharing service. The popularity of YouTube is because of ease of use and simplicity of these systems for the creation, collaboration and sharing of resources (images, videos) even from non-technical users. Current YouTube usage statistics indicate the approximate scale of the site: at the time of this writing there are more than 1 billion unique users viewing video content, watching over 6 billion hours of video each month. Also, YouTube accounts for 20% of web traffic and 10% of total internet traffic. YouTube provides many social mechanisms to judge user opinion and views about a video by means of voting, rating, favorite's, sharing and negative comments, etc. It is important to note that YouTube provides more than just video sharing; beyond uploading and viewing videos, users can subscribe to video channels and can interact with other users through comments. YouTube is generally a comprise of implicit and explicit user- user interaction. This user-to-user social aspect of YouTube (the YouTube social network) has been cited as one key differentiating factor compared to other traditional content providers. Mining the YouTube data makes more sense than any other social media websites as the contents here are closely related to the concerned topic (as it is a video content). Analyzing YouTube data using YouTube's API has become a popular trend to compare several videos as well as milking out popular videos from a country, a particular genre, a particular search term and has helped YouTube channel owners to target out their audience in a very specific manner. Once a video's analysis has begun, channels can make sure they know how the audience are reacting to their videos, in which country their videos are doing well, how high up a YouTube search result does their video come up when a particular term is being searched for etc. All this analysis has been done in this research paper where we can get to know all these details and help YouTube channel owners to grow. Providing vital information to them regarding audience responses will be a great factor in improving one's YouTube channel.

Need for Sentiment Analysis

Sentiment Analysis is an important aspect for a business owner, or, in this case, a YouTube channel owner. Social sentiment is a way of measuring the emotions behind social media mentions. It is a manner in which you can measure the tone of the conversation that's taking place—is this person satisfied, happy, angry, or annoyed? It's not enough to know that something is trending. Sentiment adds context to social media. Without sentiment, data can be easily misleading. Most social sites provide general audience metrics, but success is more dependent on actionable data that are revealed through audience participation, so it's important to comprehend positive, negative, and even neutral opinion trends. Third-party APIs are often more robust when it comes to sentiment analysis. KPIs such as conversation rate, for example, are a good place to start. The conversation rate measures the average comments per post, which tells us how engaging our posts are.

Sentiment Analysis can provide valuable insights and thus help organization's to formulate effective business strategies. It can help firms to monitor brand and product performances, handle customer grievances, get in-depth information for strategic analysis. SA can help to track and come up with effective marketing campaigns. The actual ROI of marketing campaigns can be estimated by evaluating positive and negative opinions and discussions among customers. Sentiment Analysis has become the backbone of digital strategies for most firms today. Sentiment Analysis has become the gateway to understanding consumer needs, extending customer base and expectations. It helps to pinpoint the problem and give solutions effectively. It can also help in customer segmentation by identifying segments that feel strongly about a brand or service. It can also help to identify new business opportunities. Specific phrases and texts of target audiences can be monitored to effectively generate new leads. Competitor performances can also be evaluated by monitoring mentions of the competing brands.

LITERATURE SURVEY

There has been research based on data analysis of the users, the YouTube community they are in, the people they subscribe, their subscribers, the videos they upload, the videos they watch and the category of the video watched by them [1]. Data was collected regarding the number of videos and the number of subscriber each member has in each community that was chosen from 12 categories and then the correlation factor was calculated. In their first finding they found that distribution tends to be skewed to the right. This implies that as a user uploads more videos, the probability he will get a higher number of subscriber increases. There is correlation between number of videos a user uploads and number of subscribers they get. They did graphical analysis of the same. Then the researchers conducted a network analysis where they tried to construct friend networks and subscriber networks in one group. They assumed that the users in one group should be linked tighter than linkages between users in the YouTube community as a whole.

Further studies aiming at determining the critical factors that make companies use one social media or the other according to the benefits obtained on 8 categories, from communication to increasing revenue or recruiting the most valuable employees. There is a proposed model with seven fundamental blocks of social media identity, conversations, sharing, presence, relationship, reputation and groups [2]. These blocks help determine the social media user experience and the implications for the company. They noticed that the main blocks for YouTube are sharing, conversation and reputation. They conducted surveys and studies on YouTube usability and found out that YouTube is not considered suitable for customer service, networking and recruiting but can be used for marketing purposes.

A research paper attempted to provide an overall characterization of YouTube, based on a random sample of channel and video data, by showing how video provision and consumption evolved over the course of the past 10 years [3]. It demonstrates stark contrasts between video genres in terms of channels, uploads and views, and that a vast majority of on average 85% of all views goes to a small minority of 3% of all channels. The analytical results give evidence that older channels have a significantly higher probability to garner a large viewership, but also show that there has always been a small chance for young channels to become successful quickly, depending on whether they choose their genre wisely.

They found out that there are two basic methods for collecting YouTube samples: crawling or querying the YouTube API. They tried to obtain a near unbiased data set for randomized sampling by using keyword searches which themselves can be seen as random samples. An additional filter was applied to collect only the channels with at least five uploads. In a second step, all channel and video data associated with the list from step one are retrieved via the API as specific data requests. Then they collected statistics about all the genres, different videos and channels etc.

Research papers have been written to help people in understanding the various characteristics like video length, video category, file size and bit rate, date added, views ratings, Growth Trend of Number of Views and Active Life Span of various YouTube videos [4]. It researches of the various access patterns and social networks present in YouTube.

It explains how the met-data of the videos are initially scraped from YouTube, after which the crawler extracts the information and shows us the most rated /viewed videos etc. YouTube videos form a directed graph, where each video is a node in the graph, and the crawler uses a breadth-first search to find videos related to each other in the graph.

In a research the researchers used four different techniques using machine learning to determine the popularity of the videos for on a long-term scale [5]. The 4 models being Univariate Linear (UL), Multivariate Linear (ML), radial basis functions model, Evolution Model. While one model gives priority to the daily views before the present date, the later treats all views equally. Unlike UL and ML, RBF model and the evolution model also consider the variance of the data in calculating its popularity. The researcher created a function of the views and the time – $N(v,t)$ in order to predict the popularity.

It was assumed that each video is represented as a node with a PageRank score [6]. The edges between nodes were formed to represent the relatedness of the two videos. The PageRank being the number of nodes it is connected to. This research paper examined the correlation between PageRank and multiple different statistical measures (number of views, number of comments, rate, ratings, age etc.) for a particular video. It analyzed how viewer behavior can affect the reach of the video and its influence, and how the company's advertisement can have a high reach according to the video it is linked with. Further the analysis of PageRank helped to determine connections between characteristics of videos which have high reach.

Unlike the other research papers this research paper focuses on a specific characteristic of the videos which is comments [7]. Support vector machine classification and term-based representations of comments to automatically categorize comments. It studies the relationship between comment ratings and polarizing content, more specifically tags/topics and videos, along with the analysis of distribution of comment rating amongst different video categories. According to this research, the different kind of sentiments expressed in comments, the comment ratings, and topic orientation of the discussed video content are strongly dependent on each other.

The researchers aimed to study the relationship between users and group's social networks [8]. The research concluded that both networks have a lot of similarities, like sharing the fundamental properties of most real-world networks, namely Small-Worldness and high clustering coefficient, both networks are assortative and very typical complex networks. As calculated by the researchers the degree of distribution in both cases follow the power-law. Further the average clustering coefficient and the hop distance distributions in both cases are quite identical.

Proposed Work

For Sentiment Analysis

1. Get comments from a YouTube video using its id

The page information is obtained using the request.get function with YouTube video id and the API key as parameter. The JSON file returned, is parsed with the and then appended into comments.

2. Using the training classifier to train the data-set

1. Open and read positive.txt file and store it in variable pos_sen
2. Open and read negative.txt file and store it in variable neg_sen
3. Open and read emoji.txt file and store it in variable emoji
4. Declare pos_emoji and neg_emoji as an empty list
5. Traverse through emoji.txt from i=0 till eof and declare an expression exp=""
 - 5.1 if i[len(i) -2] == '-' then run another loop 5.1.1 from j=0 till len(i)-2

5.1.1 a) $\text{exp} += i[j]$

5.1.2 Append this negative expression in as a true exp and

5.2 else

5.2.1 from $j=0$ till $\text{len}(i) - 1$

5.2.1 a) $\text{exp} += i[j]$

5.2.2 Append this positive expression in as a true exp and

5.3 Now get the extent of positive and negative response (prev and nrev) by traversing the pos_sen and neg_sen lists and finding words 'positive' and

'negative' in there respectively.

5.4 Initialize pos_set as prev + pos_emoji and neg_set as nrev + neg_emoji.

3. Getting sentiment from a sentence

Natural language toolkit i.e. nltk is used to process the comments and the comments are further tokenized.

Tokenizing the words in a sentence:

The sentences are tokenized with each word considered a token. This makes it easier for the system to compare the sentences with the classifiers and determine the negative and positive impact of the sentence.

To get positive and negative sentiment extent:

1. Declare pos = 0 and neg = 0

2. Now, in the comments list, iterate for all words which have been tokenized

3. for words in comments:

3.1 Classify the tokens and using the classifier text files, increment the pos and neg files whenever we get a match from the txt files

To get Positive and Negative Percentage:

$(\text{neg} * 100.0 / \text{length}(\text{comments}))$

(pos * 100.0 /length(comments))

To Scrape YouTube Metadata:

Scraping data obtained using an API Call using the python library pafy

1. Get all results from the API call made by inputting video id of any particular video and declare an empty list results = []
2. The data will be received as a JSON File which will be further parsed
3. Parse JSON File received and store the data in a results list declared before
4. Get meta-data and comment along with author's names in the list and write the files in scraper.csv as comma separated values.

Scraping data from top 10 videos using a search term:

1. Input the search term
2. Make an api call to the server and get the video ids of the top 10 videos which show up after searching the term in YouTube search bar
3. Now from i = 0 till i = NumberOfVideosRetrieved(in most cases = 10), iterate the video ids and run api_call_video_id() till all video ids have been scraped.

Effectiveness of the Algorithms used:

- Average time taken to get a YouTube video's metadata using its ID was calculated to be = 24.73s (Data taken from 20 tests)
- Average Time taken to get top 10 videos' metadata by using a search term was calculated to be = 4min 17s (Data taken from 20 tests)
- Average time taken to get sentiments of the video using its id was calculated to be = 9.48s (20 videos with 100 comments taken as dataset)

The algorithms are still in refinement process of modifying the system into a better model, which will have a reduced total retrieval time. A parallelized algorithm will result in time which will be equal to time taken to get data from 1 video as all processes will be run in parallel.

Implementation:

Code:

Top 10

```
from googleapiclient.discovery import build
from googleapiclient.errors import HttpError
from oauth2client.tools import argparser
import pafy
import csv

import sys reload(sys)
sys.setdefaultencoding('utf8')

DEVELOPER_KEY = "AIzaSyCi3kCoeajepzTWUUUH6jexrkWrc8w5wkg"
YOUTUBE_API_SERVICE_NAME = "youtube"
YOUTUBE_API_VERSION = "v3"
pafy.set_api_key("AIzaSyCi3kCoeajepzTWUUUH6jexrkWrc8w5wkg")

def add_data(vID,title,description,author,published,viewcount,
duration, likes, dislikes,rating,category,comments):
    data = [vID,title,description,author,published,viewcount,
duration, likes, dislikes,rating,category,comments]
    with open("scraper.csv", "a") as fp:
        wr = csv.writer(fp, dialect='excel')
        wr.writerow(data)

youtube = build(YOUTUBE_API_SERVICE_NAME,
YOUTUBE_API_VERSION,developerKey=DEVELOPER_KEY)

def get_data(videoId):
    url = "https://www.youtube.com/watch?v=" + videoId
    #Request fro Metadata of the Video
    video = pafy.new(url)

    #Request for Comments
    results = youtube.commentThreads().list(
        part="snippet",
        maxResults=100,
        videoId=videoId,
        textFormat="plainText"
    ).execute()
    totalResults = 0
    totalResults = int(results["pageInfo"]["totalResults"])
    count = 0
```

```

nextPageToken = ''
comments = []
further = True
first = True
while further:
    halt = False
    if first == False:
        print "."
    try:
        results = youtube.commentThreads().list(
            part="snippet",
            maxResults=100,
            videoId=videoId,
            textFormat="plainText",
            pageToken=nextPageToken
        ).execute()
        totalResults =
int(results["pageInfo"]["totalResults"])
    except HttpError, e:
        print "An HTTP error %d occurred:\n%s" %
(e.resp.status, e.content)
        halt = True
    if halt == False:
        count += totalResults
        for item in results["items"]:
            comment = item["snippet"]["topLevelComment"]
            author =
comment["snippet"]["authorDisplayName"]
            text = comment["snippet"]["textDisplay"]
            comments.append([author,text])
        if totalResults < 100:
            further = False
            first = False
        else:
            further = True
            first = False
            try:
                nextPageToken =
results["nextPageToken"]

            except KeyError, e:
                print "An KeyError error
occurred: %s" % (e)

                further = False

# Adding the full data to CSV

```

```

        add_data(videoId,video.title,video.description,video.author
,video.published,video.viewcount, video.duration, video.likes,
video.dislikes,video.rating,video.category,comments)

searchTerm = raw_input("Term you want to Search : \n")
search_response = youtube.search().list(
    q=searchTerm,
    part="id,snippet",
    maxResults=30
).execute()
count = 0
for search_result in search_response.get("items", []):
    if search_result["id"]["kind"] == "youtube#video":
        if count < 10:
            vID = search_result["id"]["videoId"]
            get_data(vID)
            count
            += 1
        else:
            break
    else:
        continue

```

Output:

```

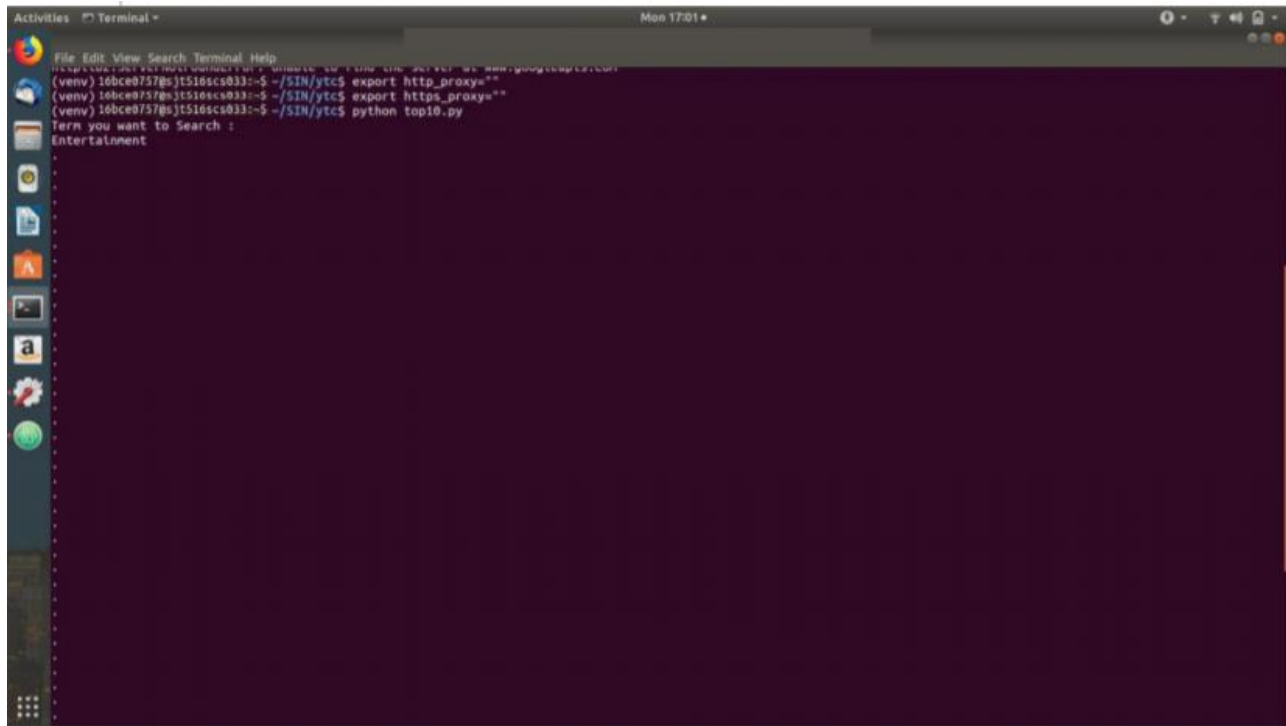
Mon 17:03
scraper.csv -- /SIN/ytc -- Atom

Project
  ytc
    .venv
    menu
    .venv
    _DS_Store
    _idsrape.py
    _scraper.csv
    _top10.py
    _DS_Store
    idsrape.py
    scraper.csv
    top10.py
  YouTube-Sentiment-Analysis
    .git
    CommentSentiment
    __pycache__
    .venv
    classifier.pickle
    comment_extract.py
    comment_extract.pyc
    driver.py
    emoji.txt
    negative.txt
    positive.txt
    progress_bar.py
    progress_bar.pyc
    sentimentYouTube.py
    sentimentYouTube.pyc
    training_classifier.py

comment_extract.py  idsrape.py  scraper.csv  positive.txt

160 Shqiptar \U0001f602'], [u'naim latifi', u'emisioni super pom doket ,aman mos lini vend per do korruptera te Kc
161 kush mendon luana eshte hilacaka le te jap like'], [u'young Boy', u'7:56 si u trem ajo gruja prej feros\U0001f
162 u'1:25:38\U0001f602\U0001f602'], [u'ilda xoxo', u'1:15:38 \U0001f602\U0001f602'], [u'EVIL GAMER', u'GANG GANG
163 jeni emisioni me i bukur ne bote qe kam pare diku\nSiper bukuri \nSi per lojra\nHjithcka duket e bukur te dua
164 \U0001f917\U0001f917\U0001f44d'], [u'EVA Ev', u'O luana te lutem mos e shfaq kaq shum finfin .. leje t kendoj c
165 [u'Mihrije Krasniqi', u'Luana je ma e bukra edhe sje mene madhe per qita te du shum'], [u'Meti ademi', u'Maesi
166 fero i ka shit drut'], [u'Kida l', u'Sa dua t martohe me kte feron\U0001f62d\U0001f602\U0001f602\U0001f60d'],
167 qart fero HAHAAHAGA mka lujt krejt'], [u'Medical Maniac', u'Un kurr nuk I kam rene pianos po jam I sigurt se
168 qellon'], [u'Youtube Meme', u'30:32 HAHAAHAAHAAHAA Momenti me gallat'], [u'Griseld Dina', u'Tha do plasi kjo t
169 argetu real . Fallco komplet . Nese doni ta vertetoni shikoni n\xeb min 36:07'], [u'Kristian Nenov T-RAW', u'
170 babaaaaa'], [u'Nix Blank', u'Like kush mendon qe fero esht me i miri'], [u'Joana Bajrami', u'Po ju prisnim \U
171 22HXTrqn468, Friends - All the states in six minutes or Chandler's dumb states game, "Friends Season 7 Episode 8
172 If You Want To Support This Channel:
173
174 https://www.paypal.com/cgi-bin/webscr?cmd=_s-xclick&hosted_button_id=S7BFCAL48CWAG
175
176 Bitcoin: 1AoXKg3d7QdUD7V6zPxFsBfn4YhNpCcnZq
177
178 Thanks for watching.", Favorite Videos, 2016-10-16 03:55:13,, 00:04:21, 23179, 466, 4.92183160782, Entertainment, "[[u
179 small ones on the eastern coast I never would have recalled though. Feels bad.'], [u'aca pavlovic', u'Or in si
180 waking up is soooo me"], [u'Philbert Chow', u'What if they got the states correctly but just numbered their f
181 forgot 10 states"" Definitely still thinks there are 56 states\U0001f602'], [u'Bluemgwes', u'""Uhhhhh, Magella
182 Barry', u'Actually when i played this i forgot 10 states lol'], [u'Ankush Nagpure', u'Ross gives the best expi
183 cares about the Dakotas!\n\nand Joey says: First of all , Utah? dude you can\U2019t just make stuff up!\n\n-
184 \U0001f602\U0001f602\U0001f602'], [u'Cailean Morrison', u'Well educated? Ross out of all the friends I would c
185 \U0001f602 that caught me off guard nd died laughing'], [u'"KRYSTLE D'SOUZA"', u'England* *is* *my* _city_'],
186 u'roos at 3:35 so so so so funny'], [u'apurva pandhi', u'3:35 hahahhahahahhahahaha'], [u'Jason Tun', u'The 4
187 of well-educated adults any Joey'], [u'Grave Gauze', u'Alabama\nAlaska\nArizona\nArkansas\nCalifornia\nColorad

```



Id Scraper

Code:

```
from googleapiclient.discovery import build
from googleapiclient.errors import HttpError
from oauth2client.tools import argparser
import pafy
import csv

import sys reload(sys)
sys.setdefaultencoding('utf8')

DEVELOPER_KEY = "AIzaSyCi3kCoeajepzTWUUUH6jexrkwRc8w5wkg"
YOUTUBE_API_SERVICE_NAME = "youtube"
YOUTUBE_API_VERSION = "v3"
pafy.set_api_key("AIzaSyCi3kCoeajepzTWUUUH6jexrkwRc8w5wkg")

def add_data(vID,title,description,author,published,viewcount,
duration, likes, dislikes,rating,category,comments):
    data = [vID,title,description,author,published,viewcount,
duration, likes, dislikes,rating,category,comments]
    with open("scraper.csv", "a") as fp:
```

```

        wr = csv.writer(fp, dialect='excel')
        wr.writerow(data)

youtube = build(YOUTUBE_API_SERVICE_NAME,
YOUTUBE_API_VERSION, developerKey=DEVELOPER_KEY)

videoId = raw_input("ID of youtube video : \n")
url = "https://www.youtube.com/watch?v=" + videoId
#Request fro Metadata of the Video
video = pafy.new(url)

#Request for Comments
results = youtube.commentThreads().list(
    part="snippet",
    maxResults=100,
    videoId=videoId,
    textFormat="plainText"
).execute()
totalResults = 0
totalResults = int(results["pageInfo"]["totalResults"])
count = 0
nextPageToken = ''
comments = []
further = True
first = True
while further:
    halt = False
    if first == False:
        print (".")
        try:
            results = youtube.commentThreads().list(
                part="snippet",
                maxResults=100,
                videoId=videoId,
                textFormat="plainText",
                pageToken=nextPageToken
            ).execute()
            totalResults =
int(results["pageInfo"]["totalResults"])
        except HttpError, e:
            print "An HTTP error %d occurred:\n%s" %
(e.resp.status, e.content)
            halt = True
    if halt == False:
        count += totalResults
        for item in results["items"]:
            comment = item["snippet"]["topLevelComment"]

```



```

        author = comment["snippet"]["authorDisplayName"]
        text = comment["snippet"]["textDisplay"]
        comments.append([author,text])
    if totalResults < 100:
        further = False
        first = False
    else:
        further = True
        first = False
    try:
        nextPageToken = results["nextPageToken"]
    except KeyError, e:
        print "An KeyError error occurred: %s" % (e)
        further = False

```

```

# Adding the full data to CSV
add_data(videoId,video.title,video.description,video.author,video.published,video.viewcount, video.duration, video.likes, video.dislikes,video.rating,video.category,comments)

```

Output:

```

(venv)16bce0757@sjt516scs033:~$ ./SIN/ytcs python ldscraper.py
python: can't open file 'ldscraper.py': [Errno 2] No such file or directory
(venv)16bce0757@sjt516scs033:~$ ./SIN/ytcs python ldscraper.py
ID of youtube video :
22HXTrqn468
(venv)16bce0757@sjt516scs033:~$ ./SIN/ytcs

```

```

1 uyPvcrRfK8, Friends - Fun Aunt Rachel, "Friends Season 7 Episode 16 ""The One with the Truth About London""
2
3 If You Want To Support This Channel:
4
5 https://www.paypal.com/cgi-bin/webscr?cmd=_s-xclick&hosted_button_id=S7BFCAL48CWAG
6
7 Bitcoin: 1AoXKg3d7QdUD7V6zxPfsBfn4YHnpCcnZq
8
9 Thanks for watching., Favorite Videos, 2016-10-22 12:14:32, 2996520, 00:04:55, 19110, 368, 4.92465209961, Entertainer
10 McGrane', u'Any Riverdale fans????!!!', [u'Mohamed Bukhary Ibn Mohamed Riyal', u'Hey , is that kid who acted z
11 audio and visual cancer. I don't even want to know why Rachel was in the bathroom teaching her nephew pee-prank
12 and daddy were not on a break.\nRach: Very good', [u'Styles Clash', u'That kid is Zack from Zack & Cody', [u']
13 Boublat', u'Carol lesbian... lol', [u'Salman Aajam', u'Which episode??', [u'Rivika Arya', u'Carol... Lesbian?
14 go to you ??\nwhat does that mean?', [u'Chummy Ngen', u'It\u2019s not in this but when Ross asked if there was
15 rude in the US or something? Oo', [u'Jayvee Nang', u'From Large to Medium to Small, Rachels boobies really are
16 break""'], [u'S Home Movies', u'BIG DADDY', [u'Surgeon', u'Get that kid an iPad...'], [u'Jenna Spagz', u'what
17 Raja', u'Now you have to throw the whole kid away.', [u'jaffa', u'""But you\u2019re not anymore "" spoken just lik
18 u'Cute Aunt Rachel :)', [u'Rocks18', u'Holy shit is that Cole Sprouse? I never noticed', [u'aristides s', u'
19 outfit as the painting of the girl in the apartment \U0001f602\U0001f602\U0001f602\U0001f602\U0001f602 min 3:5
20 \U0001f917\U0001f60a', [u'Nevaeh Grant', u'Baby jughead', [u'caitlyn who r u', u'Sucks that we never saw Ben
21 time\U0001f602\U0001f60d', [u'Blue John', u'Can\u2019t believe that\u2019s Cole Sprouse! I keep forgetting he\
22 [u'Derek McAdam', u'Cole said \u201cIt\u2019s hard to act with her cuz she\u2019s hot \U0001f525 \u201c lol ce
23 [u'Stephen Belton', u'Ross needs to learn how to have fun', [u'Poonam Poonam', u'bekar', [u'Tabitha Jaade', u'
24 serious', [u'Celestine M.', u""I love that its the kid from Big Daddy!!\nProps to anyone who's seen that movie
25 [u'Dauntingpath995', u'Saran wrap on the toilet seat so the Pee goes everywhere', [u'Kyla Lartigue', u""Omg th
26 A", u'AHHHHHH JUGHEAD JONES', [u'carter epstein', u'I need a video of Cole reacting to this ASAP', [u'\u0041c
27 who asked Ross if he ever washed his face seems to be the same student who fell in love with Ross', [u'Modroze
28 any?(tell me in comments)\nAlso what was he in first big daddy or friends?', [u'Zizi Mansour', u'I was your de
29 Cole ?! OMG (maybe it's Dylan oops lol)""], [u'Sherin Elizabeth', u'Can't just imagine how much cole has grown f

```

Sentiment Analysis

Code:

```
import training_classifier as tcl
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import os.path
import pickle
from statistics import mode
from nltk.classify import ClassifierI
from nltk.metrics import BigramAssocMeasures
from nltk.collocations import BigramCollocationFinder as BCF
import itertools
from nltk.classify import NaiveBayesClassifier

def features(words):
    temp = word_tokenize(words)

    words = [temp[0]]
    for i in range(1, len(temp)):
        if(temp[i] != temp[i-1]):
            words.append(temp[i])

    scoreF = BigramAssocMeasures.chi_sq

    #bigram count
    n = 150

    bigrams = BCF.from_words(words).nbest(scoreF, n)

    return dict([word,True] for word in itertools.chain(words,
bigrams))

class VoteClassifier(ClassifierI):
    def __init__(self, *classifiers):
        self.__classifiers = classifiers

    def classify(self, comments):
        votes = []
        for c in self.__classifiers:
            v = c.classify(comments)
            votes.append(v)
        con = mode(votes)

        choice_votes = votes.count(mode(votes))
        conf = (1.0 * choice_votes) / len(votes)
```

```

        return con, conf

def sentiment(comments):

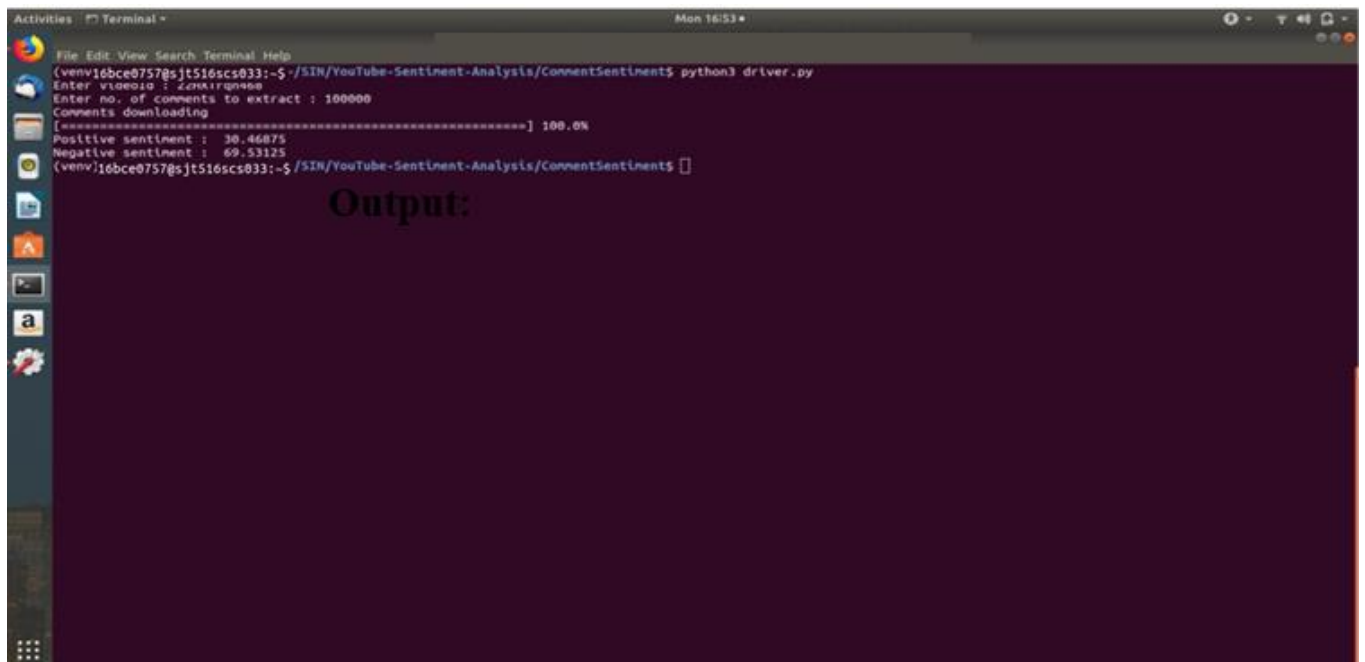
    if not os.path.isfile('classifier.pickle'):
        tcl.training()

    fl = open('classifier.pickle', 'rb')
    classifier = pickle.load(fl)
    fl.close()

    pos = 0
    neg = 0
    for words in comments:
        comment = features(words)
        sentiment_value, confidence =
VoteClassifier(classifier).classify(comment)
        if sentiment_value == 'positive':# and confidence *
100 >= 60:
            pos += 1
        else:
            neg += 1

    print ("Positive sentiment : ", (pos * 100.0
/len(comments)) )
    print ("Negative sentiment : ", (neg * 100.0
/len(comments)) )

```



The screenshot shows a terminal window with the following output:

```

(venv16bce0757@gsjts16scs033:~$ ./SIN/YouTube-Sentiment-Analysis/CommentSentiment$ python3 driver.py
Enter video id : zmxirunsee
Enter no. of comments to extract : 100000
Comments downloading
[*****] 100.0%
Positive sentiment : 30.46875
Negative sentiment : 69.53125
(venv16bce0757@gsjts16scs033:~$ ./SIN/YouTube-Sentiment-Analysis/CommentSentiment$

```

Below the terminal output, the word "Output:" is written in a large, bold, black font.

Result and Discussions

Scrapping Data based on Genre:

1. Search term: Entertainment

Time taken to display meta-data of top 10 search results: 5m 17.45s

2. Search Term: Music

Time taken to display meta-data of top 10 search results: 6m 45.10s

3. Search Term: Technology

Time taken to display meta-data of top 10 search results: 4m 56.32s

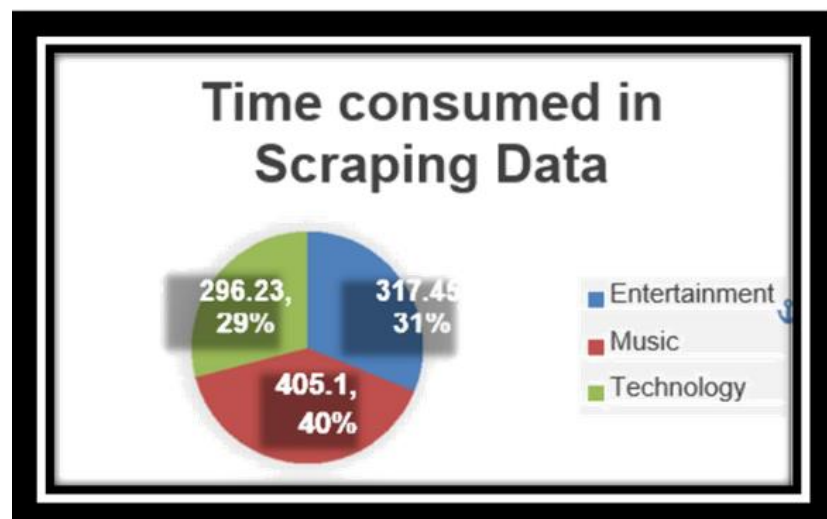


Figure 1

The above analysis shows that least time is taken to retrieve the top 10 search results of the genre, which means that out of the three it is the least popular genre as more the popularity of the genre, more will be the videos that would have to be compared.

Scrapping Data based on views:

1. Video: New iPad Pro, MacBook Air & Mac Mini Announced!

Video Id: zEx5ZQ9ir6s

Views at the time of scraping: 600k

Time taken to scrape meta-data and comments: 23.52s

Sentiment Analysis: (min. 1000 comments) Positive Sentiment: 72.3%

Negative Sentiment: 27.7%

2. Video: Empoli 1-2 Juventus | Ronaldo Double In Juve Comeback Win | SerieA

Video id: Dy3NK9WYCA8

Views at the time of scraping: 10M

Time taken to scrape meta-data and comments: 54.15s

Sentiment Analysis: (min. 1000 comments) Positive Sentiment: 77.9%
Negative Sentiment: 22.1%

3. Video: Post Malone, Swae Lee - Sunflower (Spider-Man: Into the Spider-Verse)

Video id: ApXoWvfEYVU

Views at the time of scraping: 32M

Time taken to scrape Meta-data and comments: 2min 29.40s

Sentiment Analysis: (min. 1000 comments) Positive Sentiment: 82.4%
Negative Sentiment: 17.6%

4. Video: Marvel Studios' Captain Marvel - Official Trailer

Video id: Z1BCujX3pw8

Views at the time of scraping: 44M

Time taken to scrape Meta-data and comments: 2min 29.40s

Sentiment Analysis: (min. 1000 comments) Positive Sentiment: 81.9%
Negative Sentiment: 18.1

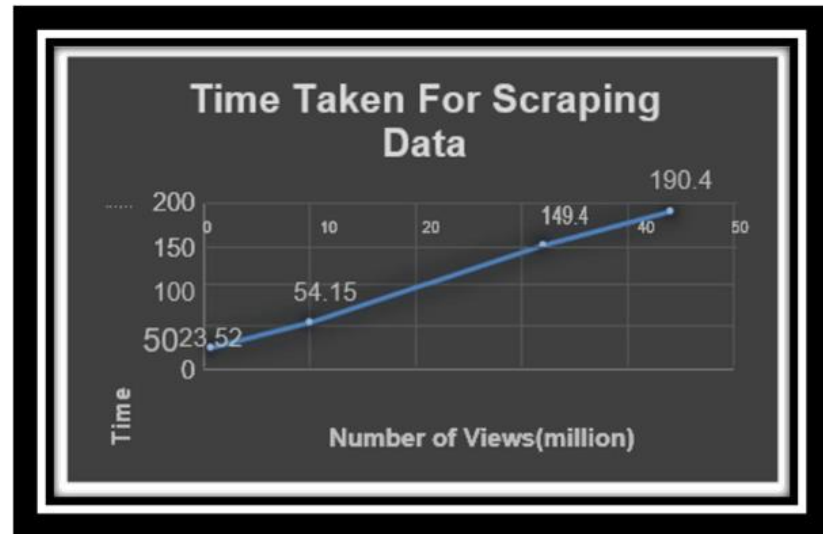


Figure 2

The above analysis portrays that the time taken for scraping of meta-data is directly proportional to the number of views on the video.

Conclusion and Future Work

This project presented a detailed investigation on analysis of various YouTube videos based on the different characteristics of videos which include the number of likes, dislikes, comments etc. This paper will help the user in filtering out the top 10 videos of various genres as well as obtaining the characteristics of a specific video. The characteristics including the number of description, likes, dislikes, comments etc. Currently all the comments are taken into account when it comes to scraping and retrieving the data, this makes the whole process time consuming. In future the time used can be minimized by limiting the number of comments being considered for being scraped and analyzed.

In future the analysis of the comments can be done with more precision using more precise training classifiers. These training classifiers will include the comparison of phrases as well, which will minimize the chances of errors. The training classifiers can further be used to train the program to not include the lyrics or the content being spoken in the video.

REFERENCES

- [1] YouTube Social Network Analysis, Authors- Ping You, Min Hu, Nayeoung Kim.
- [2] Factors influencing social networks use for business: Twitter and YouTube Authors-Alexandra Ioanid and Cezar Scarlet
<https://doi.org/10.1016/j.proeng.2017.02.496> 10th International Conference Interdisciplinary in Engineering, INTER_ENG 2016
- [3] YouTube channels, uploads and views: A statistical analysis of the past 10 years - Mathias Bartl Convergence: The International Journal of Research into New Media Technologies 2018, Vol. 24(1)
16–32 The Author(s) 2017 Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/1354856517736979 journals.sagepub.com/home/con
- [4] Understanding the Characteristics of Internet Short Video Sharing: YouTube as a Case Study Authors -Xu Cheng, Cameron Dale, Jiangchuan Liu
- [5] Predict the Popularity of YouTube Videos Using Early View Data
Author-
Anonymous
- [6] Youtube Graph Network Model and Analysis Authors -Yonghyun Ro, Han Lee, and Dennis Won
- [7] How Useful are Your Comments? Analyzing and Predicting YouTube Comments and Comment Ratings Authors- Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, Jose San Pedro
- [8] User and group networks on YouTube: A comparative analysis Authors - Malek Jebablit, Hocine Cherifit, Chantal Cherifi, and Atef Hamouda