YOUTUBE VIDEO RECOMMENDATION BASED ON USER COMMENTS AND ITS STATISTICAL ANALYSIS

[1]Akshay N. Jadhav, [2]Utkarsha A. Patil, [3]Aishwarya D. Jain, [4]Vishal L. Satpute.
[1]Graduate Student,APCOER,University of Pune, [2] Graduate Student,APCOER,University of Pune,
[3] Graduate Student,APCOER,University of Pune, [4] Graduate Student,APCOER,University of Pune
[2]Department of Computer Engineering,
[1]APCOER, Pune, India.

Abstract : YouTube is most popular video sharing platform around the world due to which YouTube has become most preferred choice of users. With the current level of complexity of YouTube ,obtaining users behavior and choices automatically became a crucial task. Current personalize recommendation system is based on users watch and search history is not adequate factor for most appropriate video suggestion. To minimize this issue of irrelevant recommendation,we are proposing the YouTube recommendation system based network of user comments and their sentiment analysis.Depending on users comments they will be added into only relevant recommendation network.We are considering that comments might be useful source to gain information about video quality and relevancy. Therefore,in this system we are using sentiment analysis approach for relevant video recommendation. YouTube dataset contains different attributes such as likes,dislikes,comments and views which can provide useful insights to the uploader using statistical analysis. We are also interested in determining the change in rate of recommendation by using our improvised approach rather than the conventional recommendation of YouTube.

Keywords:YouTube,SentimentAnalysis,Statistical Analysis,Network,Comment,Recommendation.

I.INTRODUCTION

YouTube is the most popular open platform and widely used video-sharing platform where any registered user can upload, view, comment, like or dislike any video contents. Registered users are permitted to upload an unlimited number of videos and add comments to videos. YouTube's enormous repository of video information has the potential to contain videos of interest for many users. The downside to the quantity of videos is that exploration and discovery of new, interesting, videos becomes a daunting task. So, recommendation of appropriate video becomes difficult.

The process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral. For a recommender system, sentiment analysis has been proven to be a valuable technique. A recommender system aims to predict the preference to an item of a target user. In many social networking services or e-commerce websites, users can provide text review, comment or feedback to the items. It is useful in social media monitoring to automatically characterize the overall feeling or mood of consumers as reflected in social media toward a specific application and determine whether they are viewed positively or negatively on the web. Here we are using YouTube comments to analyze the videos and for recommendation purpose.

Statistics is a branch of mathematics dealing with the collection, organization, analysis, interpretation, and presentation of data. YouTube dataset contains different attributes such as likes, dislikes, comments and views which can provide useful insights to the uploader using statistical analysis. We can provide different graphs and bar plots which will be useful for uploader to know about his overall success of the video.

II.RELATED WORK

Paolillo[8] studied friendship and its correlation to tags used to uploaded videos to the YouTube. There result indicate that YouTube uploader are strongly linked to others uploading similar content. They are addressing friend relations and their co-relations with tags applied to uploaded video. Besides these results, they mentioned that it is important to recognize that friendship is not the only relationship that structures YouTube interaction. Commenting is also another important one which they left as future study.

Mei et al.[9] proposed an online video recommendation system to suggests videos according to the current video being viewed without user profile. The related topic videos is determined by their textual features (tags, keywords). Videos with common textual features are related. This is the most common approach for recommender systems.

Krishnakumar[10] built an online video recommender system called Recoo. In Recoo, a profile is built for each user and data of user's interests is explicitly collected from the user. Recoo compares the collected data to similar data collected for other users and calculates a list of recommended items. The learned user profile is used to refine the recommendations made to match the user's preferences better. In Recoo , before the system could work, they must build each user a profile. This approach does not work when users do not want to create a profile, something that is quite common on the Internet.

YouTube represents one of the largest scale and most sophisticated industrial recommendation systems in existence. Their system at a high level and focus on the dramatic performance improvements brought by deep learning. YouTube Recommendations system is split according to the classic two-stage information retrieval dichotomy: first , is a deep candidate generation model and second is separate deep ranking model. They provide practical lessons and insights derived from designing, iterating and maintaining a massive recommendation system with enormous userfacing impact.

Candidate generation: During candidate generation, the enormous YouTube corpus is winnowed down to hundreds of videos tht may be relevant to the user. The predecessor to the recommender described there was a matrix factorization approach trained under rank loss . Early iterations of their neural network model mimicked this factorization behavior with shallow networks that only embedded the user's previous watches. From this perspective, their approach can be viewed as a nonlinear generalization of factorization techniques.

Ranking: The primary role of ranking is to use impression data to specialize and calibrate candidate predictions for the particular user interface. For example, a user may watch a given video with high probability generally but is unlikely to click on the specific homepage impression due to the choice of thumbnail image. During ranking, They have access to many more features describing the video and the user's relationship to the video because only a few hundred videos are being scored rather than the millions scored in candidate generation. Ranking is also crucial for ensembling different candidate sources whose scores are not directly comparable. They use a deep neural network with similar architecture as candidate generation to assign an independent score to each video impression using logistic regression . The list of videos is then sorted by this score and returned to the user. Their final ranking objective is constantly being tuned based on live A/B testing results but is generally a simple function of expected watch time per impression. Ranking by click-through rate often promotes deceptive videos that the user does not complete ("clickbait") whereas watch time better captures engagement.

III.PROPOSED SYSTEM

In this preliminary work, we are proposing the construction of a YouTube recommender system which will harness the user comments to create recommendation networks. This networks will help in improving the current recommendation system.

3.1.YouTube:

Using YouTube website we can watch various videos and comment on them. As YouTube is the open source platform we get data from YouTube website using API key. YouTube data API is an Application Programming Interface which allows us to get the YouTube Channel data using GET HTTP method[2]. Data, in which we are interested, is composed of a number of videos and comments on each video.

3.2.Video data:

Data from YouTube database using API key will be stored in the form of .csv file .This .csv file contains various data columns such as username, comments, comment id, reply count ,like count, etc.
Which will be useful for network formation purpose.

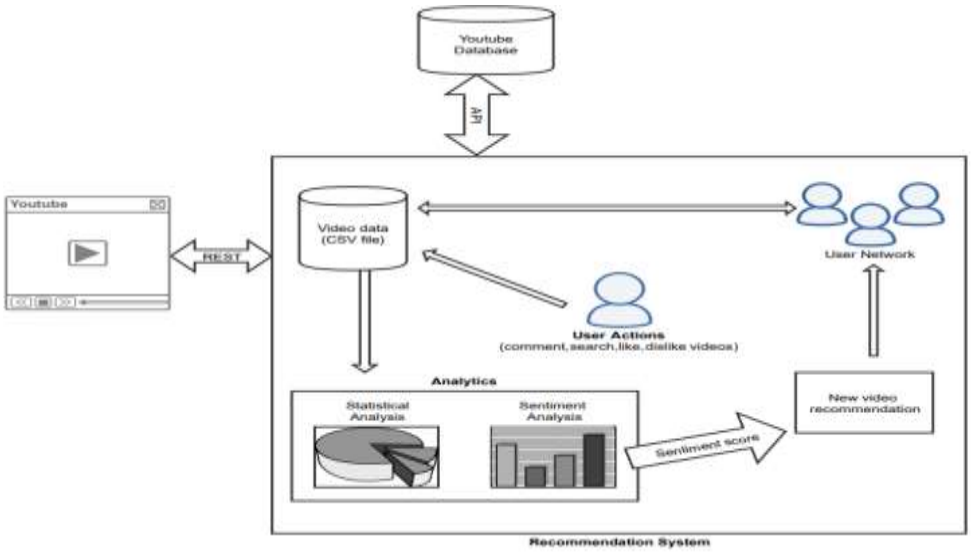

Figure1.System Architecture.

3.3.User Network:

Here, the nodes in the network represent the users, where edges are established between two nodes if a user comments on a video which is uploaded by another user. Our work is emphasizing on recommending a video to all the member nodes of a network if

any of the member is watching a video similar to the one for which the network was formed. This is possible because since the member users are part of the same network, it can be inferred that they will be commonly interested in watching similar videos. An anonymous video being watched by a user can be recommended only if this user is part of the recommendation network, which is only possible by either commenting or replying on a video owned by some other user.

3.4.Analysis:

3.4.1.sentiment analysis:
Once a member user belonging to any network starts watching any video, the recommendation system will determine the exact sentiment of users who have already seen this video and commented on it. There will definitely be a set of sentiments derived after analyzing the comments. Now, the recommendation system will make a call on whether to recommend this video to other members or not, based on how positive the user sentiments are for this video.

3.4.2. statistical analysis:
YouTube dataset contains different attributes such as likes,dislikes,comments and views which can provide useful insights to the uploader using statistical analysis.

3.5.We are using following algorithms ,libraries and its functions:

3.5.1. NRC's Extractor:
NRC's Extractor (patent pending) takes a document as input and generates a list of keyphrases as output. The algorithm uses supervised learning from examples. Extractor was trained using the same documents as we used with Eric Brill's Tagger and Verity's Search 97. Extractor is intended to emulate human-generated keyphrases. On most hardware platforms, Extractor can process a typical document in about one second.

3.5.2. get_sentences:
After loading the package (library(syuzhet)), you begin by parsing a text into a vector of sentences. For this you will utilize the get_sentences() function which implements the openNLP sentence tokenizer. The get_sentences() function includes an argument that determines how to handle quoted text. By default, quotes are stripped out before sentence parsing.

3.5.3. get_text_as_string:
he get_text_as_string function is useful if you wish to load a larger file. The function takes a single pathargument pointing to either a file on your local drive or a URL. In this example, we will load the Project Gutenberg version of James Joyce's Portrait of the Artist as a Young Man from a URL.

3.5.4. get_tokens :
The get_tokens function allows you to tokenize by words instead of sentences. You can enter a custom regular expression for defining word boundaries. By default, the function uses the "\W" regex to identify word boundaries. Note that "\W" does not remove underscores.

3.5.5.get_sentiment:
After you have collected the sentences or word tokens from a text into a vector, you will send them to the get_sentiment function which will assess the sentiment of each word or sentence. This function takes two arguments: a character vector (of sentences or words) and a "method." The method you select determines which of the four available sentiment extraction methods to employ. In the example that follows below, the "syuzhet" (default) method is called.

3.5.6. get_nrc_sentiment:
The get_nrc_sentiment implements Saif Mohammad's NRC Emotion lexicon. According to Mohammad, "the NRC emotion lexicon is a list of words and their associations with eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive)" The get_nrc_sentiment function returns a data frame in which each row represents a sentence from the original file. The columns include one for each emotion type was well as the positive or negative sentiment valence.

It is simple to view all of the emotions and their values:

Table3.1: Emotions and their values

anger	anticipation	disgust	fear	joy	sadness	surprise	Trust
0	1	0	0	0	0	0	2
0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0

Or you can examine only the positive and negative valence:

Table3.2: positive and negative valence

negative	positive
0	1
1	0
1	0
0	1
0	1
0	0

These last two columns are the ones used by the nrc method in the `get_sentiment function discussed above. To calculate a single value of positive or negative valence for each sentence, the values in the negative column are converted to negative numbers and then added to the values in the positive column, like this.

3.5.7. Collection and network analysis of social media data using vosonSML library:

The goal of the vosonSML package is to provide a suite of easy-to-use tools for collecting data from social media sources (Instagram, Facebook, Twitter, and Youtube) and generating different types of networks suited to Social Network Analysis (SNA) and text analytics. It offers tools to create unimodal, multimodal, semantic, and dynamic networks. It draws on excellent packages such as twitteR, instaR, Rfacebook, and igraph in order to provide an integrated 'work flow' for collecting different types of social media data and creating different types of networks out of these data. Creating networks from social media data is often non-trivial and time consuming. This package simplifies such tasks so users can focus on analysis.

3.5.7.1.AuthenticateWithYoutubeAPI { apiKeyYoutube}:

It is used for YouTube API Authentication..In order to collect data from YouTube, the user must first authenticate with Google's Application Programming Interface (API). Users can obtain a Google Developer API key at: https://console.developers.google.com

3.5.7.2 CollectDataYoutube(videoIDs, apiKeyYoutube, verbose, writeToFile, maxComments):

This function collects YouTube comments data for one or more YouTube videos. It structures the data into a data frame of class dataSource.youtube, ready for creating networks for further analysis.
CollectDataYoutube collects public comments from YouTube videos, using the YouTube API.
videoIDs=character vector, specifying one or more YouTube video IDs. For example, if the video URL is 'https://www.youtube.com/watch?v=W2GZFeYGU3s', then use videoIDs='W2GZFeYGU3s'. For multiple videos, the function GetYoutubeVideoIDs can be used to create a vector object suitable as input for videoIDs.
apiKeyYoutube=character string, specifying the Google Developer API Key used for authentication.
verbose =logical. If TRUE then this function will output runtime information to the console as it computes. Useful diagnostic tool for long computations. Default is FALSE.
writeToFile=logical. If TRUE then the data is saved to file in current working directory (CSV format), with filename denoting current system time. Default is FALSE.
maxComments= numeric integer, specifying how many 'top-level' comments to collect from each video. This value *does not* take into account 'reply' comments (i.e. replies to top-level comments), therefore the total number of comments collected may be higher than maxComments. By default this function attempts to collect all comments.

IV.COMPARATIVE STUDY

Table4.1:Comparative Study

System / **Charateristics**	**structure and network in the YouTube core**	**Videoreah:an online video recommendation system.**	**Recoo:A Recommendation System for YouTube RSS Feeds**	**A Recommender System for YouTube Based on its Network of Reviewers**	**Proposed System**
Platform	R Studio	Creating .idf file and database	RSS(Really Simple Syndication)	Network Workbench and Database	R Studio and Database
Security	High	High	High	Medium	High

Recommendation Based	Uploaded VideoTag (Friend and their co-relation)	CurrentVideo (history)	RSS feeds	Reviewers (Formation of Videos Network)	Comments(Formation of User Network)
Accuracy	Low	Low	High	Medium	Medium

V.CONCLUSION AND FUTURE WORK

Based on YouTube video recommendation based on user comments and its statistical analysis ,the video recommendation system can be constructed via formation of user comment's network using sentiment analysis. The user network will be form based on the data extracted from YouTube videos and by using sentiment analysis only videos with positive score will be recommended. Statistical data will be provided to the uploader.

We are considering a small network due to the limitations of the YouTube API. But, we are also currently looking at the possibility of using other datasets and augmenting them with the information about the IDs of users who wrote comments. A promising dataset is the one made available by Cha et al[6] .which contains more than 400,000 videos. Evaluation is also an important issue that we will focus on while we actually implement this recommendation system. When dealing with recommendations we should evaluate if our recommendation is effective enough. We are currently looking into this issue to see if we can devise a method to evaluate the effectiveness of our recommendation system. We would ideally like to have a metric that could be quantified to see how many users from a particular network actually watched the recommended video and find out without having to ask users directly whether they liked the recommendation or not. It will also be very interesting to analyze and compare the views of a video before being recommended and those after it is recommended to a network. Recommendation is now being made to specific network to which the viewer belongs in future.We are planning to recommend the same resultant video to other network having same interest.

REFERENCES

[1]Rahim, M. S.,Chowdhury, A. Z. M. E., Islam, M. A., and Islam, M. R. (2017). Mining trailers data from youtube for predicting gross income of movies. 2017 IEEE Region 10 Humanitarian Technology Conference (R10HTC).

[2] Bhuiyan, H., Ara, J., Bardhan, R. and Islam, M. R. (2017). Retrieving YouTube video by sentiment analysis on user comment. 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA).

[3] Paul Covington, Jay Adams, Emre Sargin. Deep Neural Networks for YouTube Recommendations. Published2016inRecSys.

[4] Muhammad Zubair Asghar, Shakeel Ahmad, Afsana Marwat, Fazal Masud Kundi.Sentiment Analysis on YouTube: A Brief Survey.2015.

[5] Choudhury, Smitashree and Breslin, John G. (2010). User sentiment detection: a YouTube use case. In: The 21st National Conference on Artificial Intelligence and Cognitive Science, 30 Aug - 1 Sep 2010, Galway, Ireland.

[6] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, I Tube, YouTube, Everybody Tubes: Analyzing the Worlds Largest User Generated Content Video System, in ACM Internet Measurement Conference, October 2007.

[7] K. Mouthami, K. N. Devi and V. M. Bhaskaran, "Sentiment analysis and classification based on textual reviews," *2013 International Conference on Information Communication and Embedded Systems (ICICES)*, Chennai, 2013, pp. 271-276.

[8] J. Paolillo, "Structure and network in the youtube core," in Hawaii International Conference on System Sciences. IEEE Computer Society, 2008, pp. 146–156.

[9] T. Mei, B. Yang, X. Hua, L. Yang, S. Yang, and S. Li, "VideoReach: an online video recommendation system," in Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2007, pp. 767–768.

[10] A. Krishnakumar, "Recoo: A Recommendation System for Youtube RSS Feeds," University of California, Santa Cruz, Tech. Rep., 2007