YouTree- A Visualization Paradigm of Statistically and Textually Similar Videos

Dhanasekar Sundararaman
Department of Information Technology
SSN College of Engineering
Anna University
+91 97911 31867
dhanasekar312213@gmail.com

Vishwanath Seshagiri
Department of CSE
College of Engineering Guindy
Anna University
+91 95000 11210
vishwa.nath@outlook.com

Balaji Ramesh
Institute of Remote Sensing
College of Engineering Guindy
Anna University
+91 88072 70640
balaji.9th@gmail.com

Priya Arora
Department of Information Technology
Department of Computer Science
Texas A & M University
Texas, USA
arora.priya4172@gmail.com

ABSTRACT
The rise of social media usage in the form of multimedia is on an exponential increase owing to the increased internet bandwidths in the recent past. As a result, people communicate in the form of videos and images a lot more than ever. One such video sharing and content developer platform is YouTube. YouTube has many features on video analytics in the form of recommendation systems, monetisation etc. It also offers many features for developers to evaluate their content and offers insights on the performance of their videos. Though these features are available, there is not even a single feature for developers to evaluate their content based on the performance of other's videos, which share the same nature of the content - the similarity between any two videos. Here, the similarity between two videos has a statistical measure apart from the content, which includes description and comments of a video. Thus, we propose an analysis of a query video and a range of videos to determine the most similar videos using statistical and textual similarity. The statistical similarity is evident from the number of derived features extracted from a video and the textual similarity is found by analysing the text data from the description and comments of a video. Experimental results show that the resultant similar videos are highly representative of both the statistical and textual similarity and can be used as a measure to compare two videos.

CCS Concepts
• **Computing Methodologies→ Artificial Intelligence → Natural Language Processing → Information Extraction.**

ICCMS 2018, January 8–10, 2018, Sydney, Australia

ACM ISBN 978-1-4503-6339-6/18/01…$15.00
https://doi.org/10.1145/3177457.3177467

Keywords
Textual Similarity; Statistical Similarity; Social Media Mining; Natural Language Processing.

1. INTRODUCTION
Information sharing has been a crucial part of our growth as a society, and since the advent of Social Media, the ease of permeating a piece of news has become more comfortable. With the increase in Internet Bandwidth, knowledge sharing has shifted from textual medium to multimedia platforms. One of the more popular platforms is YouTube, where almost five billion hours of video are watched every day across the world by more than a billion people. YouTube attracts viewers by giving unlimited access to the videos on their platform and also enables the content creators to monetise their work by displaying ads before the video. Performance of a video is evaluated using different parameters including the number of views, the target demography, and virality of content. These performance metrics engenders the creators to fabricate quality content to garner maximum views possible from the most comprehensive range of audience.

On the flip side, this incites the creators to ride the trending tide by creating videos, related to current topics in trend. For, e.g., a narrative about the history of presidents of a particular country during the presidential election, encompasses the audience, as they are looking for more information to make the right choice. The creators can check the reach of their video, through comprehensive statistics provided by YouTube and also have a check on the amount of money earned via advertisements. Even though comprehensive statistics exist there isn't a mechanism to compare the relative performance of similar videos by other channels. By relative, we mean videos which have similar content, baseline statistics and comments.

Evaluation criteria must exist for the videos, to efficiently compare them and produce right kind of recommendations for the viewers. Evaluating a video from similar videos based just on the textual similarity of description or comments can be absurd at times because of the anachronistic nature of different videos. Thus, a more comprehensive form of comparison is explored by us, in the form of Statistical Similarity, where videos are compared based on different parameters mentioned in Section 3. We explain this Statistical Similarity with great details in Section 5, while the explanation of textual similarity exists in Section 6.

To Summarise,

• Features which are publicly available are obtained for approximately 0.5M videos through the Youtube's official API.

• Using these direct features, we formulate derived features which capture the more subtle details about the videos which these direct features could not, and perform exploratory data analysis using these features.

• We then find statistically and textually similar videos for a given query video and combine them to get a final list of similar videos. This analysis of the relationships does not consider the content of the videos which makes it a relevant work to do for future. Hence, this paper provides us a robust baseline for our contribution in this field.

• We find influential words between these set of similar videos with the query video using Influential Term Metric proposed in Section 6.

• Finally, we construct YouTree, a visualization paradigm that consists of the statistical and textual similar videos for a given root query video.

2. RELATED WORKS

Youtube recommendation has attracted a lot of attention in the research community in the recent years. Song et al [1] uses information extracted from reviews made as comments on various videos to determine the relationship between videos in order to make the network. The nodes in the network represents the videoID, the weight of link between two nodes represents the number of common commenters on the two videos. However, they have not used a much larger system to evaluate their model, as a large number of difficulties occur when scaling a model. To overcome this difficulty, our recommendation system uses metadata of 0.5M videos. Not only this, we have taken the combination of features too to evaluate our system with such a large data-set which gives us a qualitative system for recommending youtube videos.

A multi-modal recommendation system by Mei et al. [2] uses textual features for building a recommendation system. This system suggests videos based upon the currently watched video. This is the most common recommendation system for not requiring a user profile. It is obvious that this system may get plenty of videos to the user but this approach does not rank the videos whereas our system creates a YouTree of the videos which helps in ranking the videos for recommendation.

Another recommendation system - Recoo built by Krishnakumar [3] provides a baseline for our work as how a robust systems for recommendation can be made. This system uses data of each user interests collected from the user. This system provides recommendation by comparing the collected data for a user with another users and hence calculates a list of recommended items for the user. This study gives us a valuable approach to build our YouTree. A domain specific (health care) recommendation system proposed by Rodriguez et al. [4] recommends videos from YouTube related to health domain. The used content is mainly taken from Medline Plus and the main factor is the involvement of medical-related videos. Though it provides a good recommendation system but only for a specific domain. However, these approaches do not work when the user does not have an account which has been taken care of in our work - YouTree.

The study done by Cheng et al. [5] for the analysis of youtube videos shows direct relation between rank and various characteristics of a video like reviews, comments, and its ratings. This study has taken few features into consideration like ratings, number of views, number of comments etc using 3M videos. This evaluation has given us a baseline to make a robust recommendation system using available as well as so many derived features.

Another work done by Davidson et al. [6] uses batch-oriented precomputation approach rather than on-demand calculation of recommendations. This causes delay between generation and serving a particular recommendation data-sets which is mitigated by continuous updation of the data-set. It cannot serve the need for an immediate recommendation, unlike our work.

Youtube has a large number of users who watch multiple videos. The correlation among these videos is measured by co-view numbers. This can also be understood as the number of same videos watched by some number of people. The same study is done by Baluja et al. [7] which plot the graph using this information. This information of co-view is used to recommend the videos for the system. However, this provides a limited overview of the recommendation system.

Some work has focussed upon graph topology and contextual information for building the system as done by Dahimene et al. [8]. The study done by the Covington et al. [9] talks about the improvement in performance of the existing Youtube recommendation system by deep learning. This system works in two-stage information retrieval manner. It also provides insights as to how this performance can be achieved and the practical aspects of it but it does not rank the videos whereas our system creates a tree of YouTube videos which helps in ranking the videos for the recommendation. However, such limited features might not predict a large number of videos in an accurate way. So, to provide such a robust and accurate system, we propose YouTree.

3. EXPLORATORY DATA ANALYSIS

The videoIDs obtained from YouTube 8M Dataset, access token was generated through Google Console and was passed to YouTube Public Data API. As shown in Figure 1 using this API, the following statistical features of videos and corresponding channels was queried by us; count of views, likes, comments, dislikes, total views of a channel, videos published by channel and time of publishing. Apart from these, textual features like comments of a video, and their descriptions are mined by us. While collecting the data, some videos were removed from YouTube after a certain period, either by the channel or by YouTube. The features of these videos were not considered for evaluation as they can no longer be suggested.

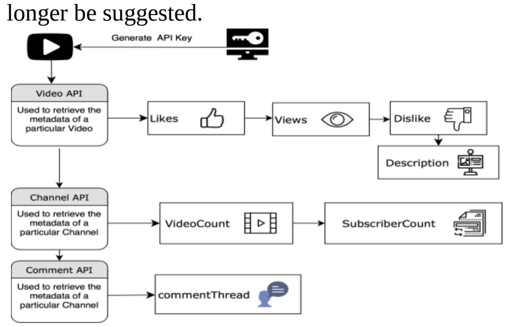

Figure 1. Process to obtain the features using YouTubeAPI.

These features give a straightforward view of the system, ignoring the methods which are used to obtain these metrics. All these features garnered a p-significance value greater than 5 percent, thus invalidating their significance. Direct Features can be increased by placing appropriate celebrities, naming the videos tactfully and by other click-baits which entice the viewer into watching the video. As a channel grows the number of individual features increases, but the overall longevity and consistency of the videos need to be analysed. Furthermore, videos of a certain category generally tend to have better overall when compared to videos belonging to minor domains. To overcome these barriers 15 derived features were extracted from the existing data and reduced to 10 features whose p-significance value was less than 5 percent. These features are; Dislikes per View, Comments per View, Views per Subscriber, Elapsed Time of a Video and Channel, Views per Elapsed Time, Total Views of Channel per Total Video Count, Comment Like and Dislike Count Per Subscriber. These features help us acquire a deeper understanding of the content network that drives the platform. Furthermore, it enables us to look past the names associated with the video, and concentrate solely on the brevity of the video, and the person posting it. As a channel grows the number of individual features increases, but the overall longevity and consistency of the videos need to be analysed.

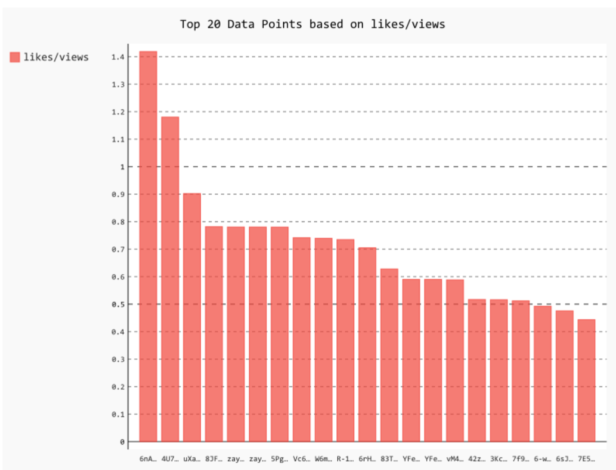

Figure 2. Top 20 videos by likes per view.

Once these features are discovered, a bar graph is plotted for number of likes per view and it is compared with the bar graph of view count of a video, as shown in Figure 2 and 3 respectively. The individual parameters of a video with similar likes is taken and compared in Table 1, to illustrate the significance of these derived features. It is found that, even though a video is widely watched by plethora of people, it didn't have content captivating enough to entice the attention of these audience. The comparison between these video statistics shows that our system recommends videos of higher degree of content, when compared to the system built by Song et al.[1]

This metric is extended for the channel, by comparing the subscriber count, with likes views and dislikes per subscriber count. These features indicate the popularity of a channel in retaining viewership, indirectly giving a holistic picture about the content created by these channels. The elapsed time of video and channel, are derived from the Published Times, and compared with the likes, views and comments garnered by these videos during this time period. This is performed in order to understand how a particular content has performed over time, these results shown via a box plot in Figure 4. As is evident, the number of videos with high views per elapsed time is extremely low, and the next value is less than half of the previous value. This indicates the need for separating content that has been consistent over long periods, over the ones who enjoy their ten seconds of fame.

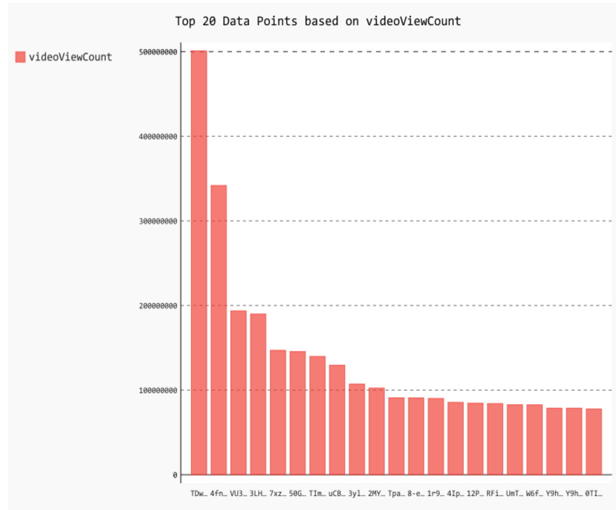

Figure 3. Top 20 Videos by View Count.

Table 1. Comparison of Videos with Similar Likes

VideoID	Likes	Views	L/V
7f90DU71fmE	403562	10,86,792	0.371333244
TDwBuPsrSM4	474357	524,773,036	0.0009039279

Most of the times the popularity of a channel also directly leads to the growth in individual features of a particular channel. To overcome this barrier, we plotted the graphs for likes per subscriber, which enabled us to isolate the channels where actual good content is being delivered. This can be seen evidently in Figure 5, as the most of the videos have a much lowers likes per subscriber ratio indicating the inconsistency of considering the individual features.

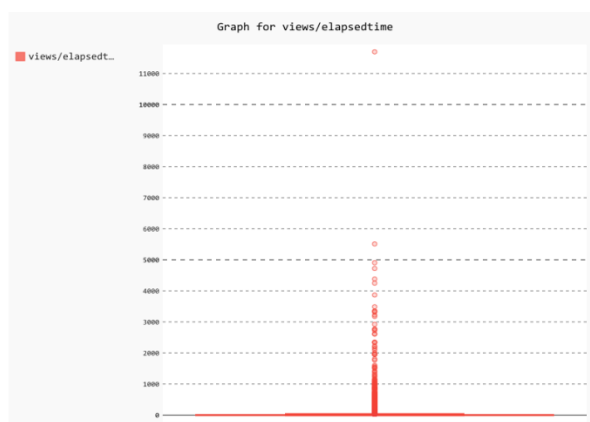

Figure 4. Box Plot for Views Per Elapsed Time.

YouTree takes into consideration longevity of a video, and consistency of a channel while finding Statistical Similarity, transitively resulting in recommendation of gnarly content. Through Exploratory Data Analysis, it can be concluded that the features derived by us, are paramount to suggesting videos which are similar to the queried video along with having remarkable content, to make sure the viewer does not get disinterested.

YouTree takes into consideration longevity of a video, and consistency of a channel while finding Statistical Similarity, transitively resulting in recommendation of gnarly content. Through Exploratory Data Analysis, it can be concluded that the features derived by us, are paramount to suggesting videos which are similar to the queried video along with having remarkable content, to make sure the viewer does not get disinterested.

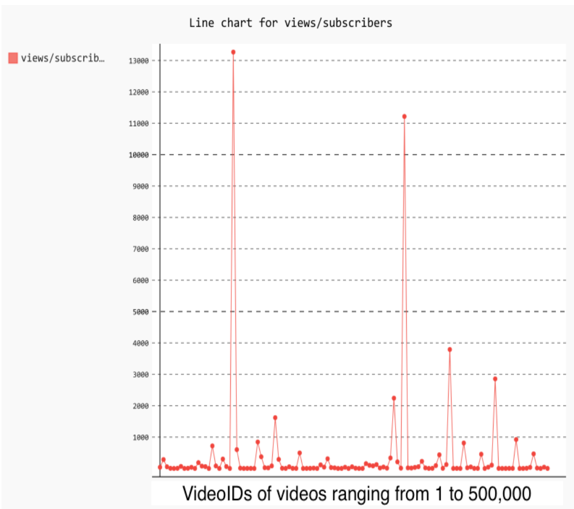

Figure 5. Line Graph for Views Per Subscriber.

4. TEXTUAL SIMILARITY

In this section, we group the comments and description of all the videos, into a single document and name the document using videoID. Grouping of textual data about a video is performed by us, to find the videos with similar text using different distance metrics. After creating the documents, the top words in each document is analysed using TF-IDF scores which gives a holistic picture of the choice of words each video uses. There are various distance metrics such as Euclidean, Manhattan, Cosine but Cosine Distance is the best fit considering the bias caused by different document lengths evident from TF-IDF Scores as the inner product of two vectors is divided by product of their vector lengths, resulting in value between 0 and 1. The cosine distance is given by,

$$\psi = (\chi 1.\chi 2)/||\chi 1||||\chi 2||$$

The similarity score is obtained by subtracting this value from

$$\alpha = 1 - (\chi 1.\chi 2)/||\chi 1||.||\chi 2||$$

While calculating the distance between queried video and other videos present in the corpus, words present in query video only are considered as the origin from which distances are calculated. The TF-IDF scores corresponding to these words is taken as baseline for calculating the cosine distance since the recommendations are made with respect to the queried video.

5. STATISTICAL SIMILARITY

Statistical Similarity is the notion of finding content similar to the one that has been queried by the user. For this method, we consider the features mentioned in Section 3, grouping them in different bins according to their overall performance. We take the top 1000 Textually Similar videos, and perform Statistical Similarity on them. The algorithm for Statistical Similarity is given below.

5.1 Statistical Similarity Algorithm Variables

Video Features is an array of size [N * F], where N is the number of Features, and F is the number of Statistical Parameters under considerations. Then the BINS are set as an array with size [F * 10], representing the 10 bins or clusters. Here, bins represent the videos which are grouped together under a banner to ease searching for similar videos as entire corpus needn't be scanned each time. Bin ID Vector Of Videos will become an array of size [N * 10], where each [1 * 10] vector will be the BIN IDs from 1 to 10.

5.2 Statistical Similarity Algorithm

1Function Get_Bins(feature, bin_intervals):

1.1. if bin_intervals isEMPTY:

1.1 bin_intervals.append(MAXIMUM(feature))

1.2. if SIZE_OF bin_intervals >= 10:

2. 1 return REVERSE_OF(bin_interval)

1.3. bin= equally spaced bin intervals 0

1.4. bin_intervals[LADT]

1.5. percentage_hist = histogram(feature, bins)

1.6. for 10 iterations of bins:

1.6.1 Add percentage in current bin until 10%

1.6.2 if percentage_of_videos >= 10:

1.6.2.1 bin_intervals.append(bin[i])

1.6.3 recurse to further divide the bin intervals

2. for i = 1 till f:

2.1 feature = video_features[ALL, i]

2.2 BINS.append(get_bins(feature, [EMPTY]))

3. bin_id_vector_OF_videos= bin number vector for each video as per the intervals in BINS

4. distances=SUM(ABS(bin_id_vector_OF_videos-bin_id_vector_OF_videos[target_video]))

We explain the algorithm as follows:

i Find the bin intervals for each attribute using equal interval.

ii. The bin intervals are created so as that each bin has bincount at most 10% but not more than 10% of the entire data.

iii. Incase any bins has bincount more than 10% then it is further subdivided into smaller bins.

iv. Then each video attribute is allocated to a bin using bin id as per the bin intervals created above.

v. The same allocation is done for all the attributes of all the videos.

vi. Find the bin id for the attributes of target video.

vii. Then the distance is calculated by taking the sum of absolute of difference between the bin ids of the target video and the comparing video

viii. The distances are calculated for all other videos versus the target videos and the similarity is given based on the minimum distance.

6. YOUTREE

Once the Textual and Statistical Similarity of videos is found in Section 4 and 5 respectively, we build YouTree using these results, to recommend videos for the user efficiently. The construction of Tree, after finding textual similarity is shown in Figure 6, and YouTree is shown in Figure 7 to efficiently compare the two trees and elucidate the novelty of our work.

In Figure 5, the user queries the video “4fndeDfaWCg” which belongs to “Backstreet Boys” song “I want it that way,” which is one of their hit songs. Upon comparing textually, the top recommendation belonged to another “Backstreet Boys” song “I Still,” released by VEVO around the same time as the queried song. The description and comments in both videos were almost similar, with many commenters voicing positive reviews about both the videos. As the tree grew vertically, each video recommended was related to the video in previous level, as well as the queried video. The videos recommended were analogous by nature, as the second layer of YouTree contained hit songs of various Pop Artists in the 1990s, and featured similar comments and description. Further down the tree, the videos suggested were similar to the previous level, but were starting to deviate from the originally queried video while retaining the genre of search.

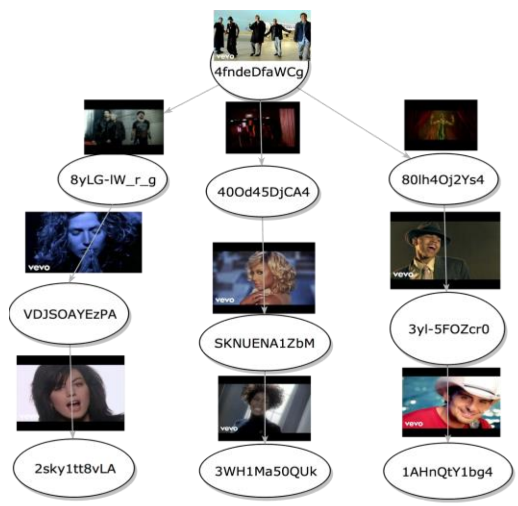

Figure 6. Tree generated after finding Textual Similarity.

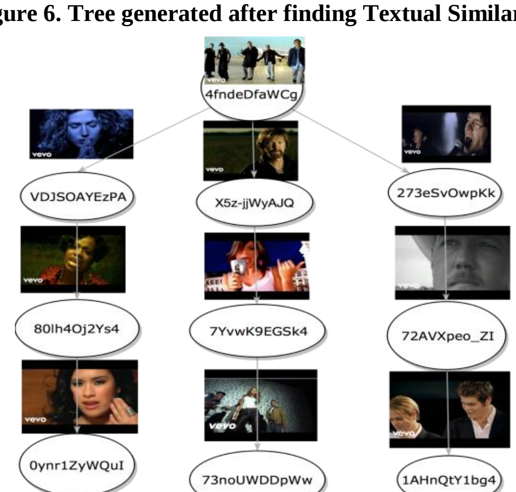

Figure 7. YouTree: A visualization.

Subsequently, the top 1000 textually similar videos are retrieved and checked for Statistical Similarity, as mentioned in Section 5 and YouTree is generated based on the results. The YouTree generated after checking the statistically similar videos is shown in Figure 7 and is akin to that of Figure 6, with improvements in the results generated. The first suggestion is that to “Sophie B Hawkins” song “Right Beside You”, which is present in 2nd layer of the tree in Figure 6. The change in layers is attributed to the striking similarity of statistical features as both videos were uploaded to YouTube on “October 25th 2009” and since then have garnered similar numbers in terms of viewership, likes and other features derived by us. Correspondingly the video with ID “80lh4Oj2Ys4” is pushed down to second layer, from first layer as the videos are less statistically similar when compared to the others. In YouTree, most of the videos suggested via Textual Similarity are not present attributing to the quality of content, determined by the individual characteristics of the video. Thus, our model present a novel look into the structuring of Multimedia Content Recommendations on a Public Platform like YouTube.

7. CONCLUSION

As stated in the abstract, we developed a visualisation paradigm to view statistical and textual similar videos for a given query video, and recommend similar videos to the viewer to inveigle the user, making sure he continues using the platform. As shown in Figure 7, the videos recommended in each level are similar to the originally queried video and presents a visualisation for the user to feel connected with the platform. Our work takes into account the current mindset of the user, by taking in a query video and recommending quality content similar to that, which is lacking in multiple works, who consider all the queries of user and the trending videos before suggesting a video. As the tree level goes deeper, we gradually deviate from the originally queried video, but retain the genre in which user is currently interested. In future, we would also like to incorporate video analytics to analyse all the frames of a particular video in comparison with other videos.

8. REFERENCES

[1] S. Qin, R. Menezes and M. Silaghi, *A Recommender System for Youtube Based on its Network of Reviewers,* 2010 IEEE Second International Conference on Social Computing, Minneapolis, MN, 2010, pp. 323-328.

[2] Tao Mei, Bo Yang, Xian-Sheng Hua, Linjun Yang, Shi-Qiang Yang, and Shipeng Li. 2007. *VideoReach: an online video recommendation system*. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval(SIGIR '07). ACM, New York, NY, USA, 767-768.

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

[4] A. Rivero-Rodr ǵuez, S. Th. Konstantinidis, C.L. Sanchez-Bocanegra, L. Fern ández-Luque, “*A health information recommender system: Enriching YouTube health videos with Medline Plus information by the use of SnomedCT terms*”, in IEEE Xplore 2013.

[5] Cheng, C. Dale, and J. Liu, *Statistics and social network of Youtube Videos,* in Proc. of IEEE IWQoS. Citeseer, 2008.

[6] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, *The YouTube Video Recommendation System,* in RecSys2010, 2010.

[7] Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, Mohamed Aly, *Video Suggestion and Discovery for YouTube: Taking*

Random Walks Through the View Graph, in WWW ACM 2008.

[8] Ryadh Dahimene, Camelia Constantin, Cédric du Mouza, RecLand*: A Recommender System for Social Networks*, in CIKM 2014.

[9] Paul Covington, Jay Adams, Emre Sargin,"*Deep Neural Networks for Youtube Recommendations"*, Research at Google, 2016.