Video Recommendation and Link prediction

Ganesh Aggarwal (2019csb1085)
Shobit Prashant Nair(2019csb1121)
Vanshal Singh (2019csb1129)

November 2020

Abstract

Link prediction and recommendation has been a common problem in social graphs. Analysing the network and recommending the next best graphical node is of major interest of all social networking sites. Unlike the majority of the articles where link prediction was done on social networks , we rather focused on OTT (Over - the - top) platforms. Eg:- Netflix , Disney+ , Amazon Prime etc:- Initially we decided to do the same on youtube network but the dataset we obtained didn't have video titles (rather had unique serial codes) due to which evaluating the results via user discretion got very difficult.However, we later found a dataset of OTT platform involving known movie titles which made us make a change from our initial decision of working with youtube. Firstly , We analysed the dataset where each video has multiple attributes like (Directors , Actors , Views , Genre , Country) tagged to it. Using this tagged data , we created a probable graph by connecting movies/shows with common attributes in a sensible and systematic manner. We start by introducing the use of Adamic adar measure and K-means clustering for improving similarity measures and finally show how our approach extends to provide recommendation given a query of movies/shows.

1 Introduction

OTT (Over-the-top) platforms are exclusive streaming companies which stream movies/shows and sometimes documentaries. The user has to pay a monthly subscription to avail the premium contents in such OTT platforms. Considering it is a paid content , the user expects much greater experience from them rather than any normal free video sharing platform like youtube. Recommendation-system is a very important aspect as maintaining the viewer interest and keeping the user glued to stream more shows. Therefore , a good recommendation system can be a great investment for the company to maintain a good long-term relationship with the user and improve customer satisfaction.Just like how facebook created a methodology for suggesting a probable friend by analysing mutual friends , similarly a robust methodology which analyses such unique attributes about a movie can be used to recommend some 5-7 other shows that user may like.

1.1 Problem

The main issue is to obtain the best 5-7 possible recommendations from a sea of other videos in the network. Some naive approaches may take exponential time to execute this task. There are many ways now to efficiently do link prediction by doing biased random walks like pixie random-walks in large data sets. However , the network scale in OTT platforms are relatively smaller compared to a network such as YouTube. Hence , it requires finesse over the way we perform the recommendation and the need to understand the network in grass-root level.

1.2 Literature

These are some reports that gave us deep insight on the idea of link prediction [Sha18] [HZ14] .
We implemented k-means clustering on movie titles and description by following the python implementation walk through from the following article [Sal18]. We used the Adamic Adar index for link prediction with the help of the following wiki. [AA]

1.3 New idea

Considering the smaller scale in size compared to other social networks , we thought of implementing a much more robust method by using the attributes of each movie and analyse which common attribute brings up better recommendation and what attributes do pivotal role in deciding how to rank the best movies the user might like. We plan on creating a whole new graph from this key idea and analysing how we can bring the best recommendations from it.

1.4 Implementation details

The programming language used is Python3 . The libraries used were Networkx , Pandas , MatplotLib and Sklearn. Networkx and Matplotlib were used for graph construction and visualisation , Pandas for reading the csv dataset efficiently and some inbuilt functions in Sklearn to implement Text clustering using K-means and Term Frequency-Inverse Document Frequency (TD-IDF)
Firstly we load the dataset which is a csv file using pandas library.

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in
81,145,628.00	Movie	Norm of the North: King Siz...	Richard Finn, Tim Maltby	Alan Marriott, Andrew Toth, Brian D...	United States, India, South Korea, ...	September 9, 2019	2,019	TV-PG	90 min	Children & Family M
80,117,401.00	Movie	Jandino: Whatever it Takes	-	Jandino Asporaat	United Kingdom	September 9, 2016	2,016	TV-MA	94 min	Stand-Up Comedy
70,234,439.00	TV Show	Transformers Prime	-	Peter Cullen, Sumalee Montano, Fr...	United States	September 8, 2018	2,013	TV-Y7-FV	1 Season	Kids' TV
80,058,654.00	TV Show	Transformers: Robots in Di...	-	Will Friedle, Darren Criss, Constanc...	United States	September 8, 2018	2,016	TV-Y7	1 Season	Kids' TV
80,125,979.00	Movie	#realityhigh	Fernando Lebrija	Nesta Cooper, Kate Walsh, John Mic...	United States	September 8, 2017	2,017	TV-14	99 min	Comedies
80,163,890.00	TV Show	Apaches	-	Alberto Ammann, Eloy Azorín, Veró...	Spain	September 8, 2017	2,016	TV-MA	1 Season	Crime TV Shows, Int
70,304,989.00	Movie	Automata	Gabe Ibáñez	Antonio Banderas, Dylan McDermo...	Bulgaria, United States, Spain, Ca...	September 8, 2017	2,014	R	110 min	International Movies
80,164,077.00	Movie	Fabrizio Copano: Solo piens...	Rodrigo Toro, Francisco Schultz	Fabrizio Copano	Chile	September 8, 2017	2,017	TV-MA	60 min	Stand-Up Comedy

Figure 1: A glimpse of the csv dataset we used

We obtain a dictionary containing row number as unique id for each video and each attribute acting as a label for individual video in the dictionary. Now , we will first implement text clustering onto the loaded data by applying K-means clustering and use TD-IDF vectorizer to rank title and description similarity of each video with all others in the dictionary. The implementation of the above is done via the help of the following article [Sal18] . The outcome after the process will give us a (k-sized)list mapped to each video sorted from most similar to least based on their similarity in titles and description.

```
def find_similar(tfidf_matrix, index, top_n=7):
    cosine_similarities = linear_kernel( tfidf_matrix[index:index+1], tfidf_matrix).flatten()
    related_docs_indices = [i for i in cosine_similarities.argsort()[::-1] if i != index]
    return [index for index in related_docs_indices][0:top_n]
```

Figure 2: The function which gives us the k-best similar videos based on common description

Now , we will move onto forming the graph using Networkx Library. Our graph has 5 type of nodes (Person[Directors , Actors] , Country , Genre , Movie , Keyword). MOVIE nodes are our main nodes and all other are attribute nodes(for simplicity , we have referred them as foci nodes).While forming the graph each 'MOVIE' node is added an edge with all the other unique foci nodes which are attributes of this movie. However , the foci type 'Keyword' is a bit different , we will take either the first 8 characters or 80 percent of the prefix (whichever is minimum) from the movie title as string and find the K most similar named movie titles and their description(Using find similar function in figure 2) corresponding to this string. The string is then formed as a new foci in graph and all the K+1 videos (1(current movie) + K(similar named movies)) are added edges to this foci node.This can be verified in subgraph figure 6 in which the orange coloured nodes denoting SIMILARITY FOCI. The movie 'Men in Black has nearly '10 edges connected to different text based similarity foci'. We will repeat this procedure for all the movies in the dataset and the final graph is formed which will have the movie nodes and the FOCI nodes (ACTORS , DIRECTORS , KEYWORD , GENRE). Now , we will take the assistance of Adamic Adar index(AAI) to bring out best from our recommendation engine. The AAI is a trivial and amazing concept to predict links in graphs. Basically , we will explore the neighborhood of the target movie (which are a list of different type of foci nodes).Exploring these foci nodes will give us a new set of movie nodes. We then add a pair [movie , common foci] for each movie into the intersection nodes dictionary used in Figure 3. Basically , in graphical perspective , it means that more the common foci nodes that two movies share , more will be their similarity measure.

Our insight into Adamic Adar measure :-

- Let x and y be the 2 nodes (Movies to be precise)
- let degree(v) mean the number of neighbors of node v and N(v) denote the set containing neighbors of v.

$$AdamicAdar(x, y) = \sum_{v \epsilon N(x) \cap N(y)} \frac{1}{log(degree(v))} \tag{1}$$

- Basically, for each node v in common to x and y, add to the measure 1/log(degree(v))
- if x and y share a node v that has a lot of adjacent nodes, then the value of the weights will be low as degree(v) will be high and will decrease net value of 1/log(degree(v)).
- if x and y share a node v that does not have a lot of adjacent node , then the value of the weights will be high as degree(v) will be low and will increase net value of 1/log(degree(v)).

```
# Adamic measure
for key, values in intersection_nodes.items():
    w = 0.0
    for e in values:
        w = w+1/math.log(G.degree(e))
    movies.append(key)
    weight.append(w)
```

Figure 3: Code snippet of Adamic Adar Measure

We have implemented it in such a manner that we obtain the sorted list based on their weights (Higher to lower) for any valid movie title(String) as query. For interpreting if our data is coming out correct , we will analyse the sub graphs involving the target node and recommended nodes.

2 Results

2.1 Experiment findings and interpretation

The results for the queries are obtained as a list containing [Movie title , Similarity index].The similarity index is the value obtained via Adamic Adar measure of the target video. Here are the Outputs for queries against few popular movie titles.

```
show("Men in Black II")

================================================================
 Recommendation Shows for : Men in Black II
================================================================
Men in Black                                        4.411281
Wild Wild West                                      1.960923
Twin Peaks                                          1.569881
Movie 43                                            1.346214
Evolution                                           1.301349
Spy Kids                                            1.179999
Small Soldiers                                      1.174500
The Dukes of Hazzard                                1.047707
Skiptrace                                           1.047707
Space Cowboys                                       1.031926
Seven Pounds                                        0.995775
Indiana Jones and the Raiders of the Lost Ark       0.972931
Rodney King                                         0.968758
Taken                                               0.968758
Bright                                              0.908110
dtype: float64
================================================================
```

Figure 4: QUERY : Men in Black II

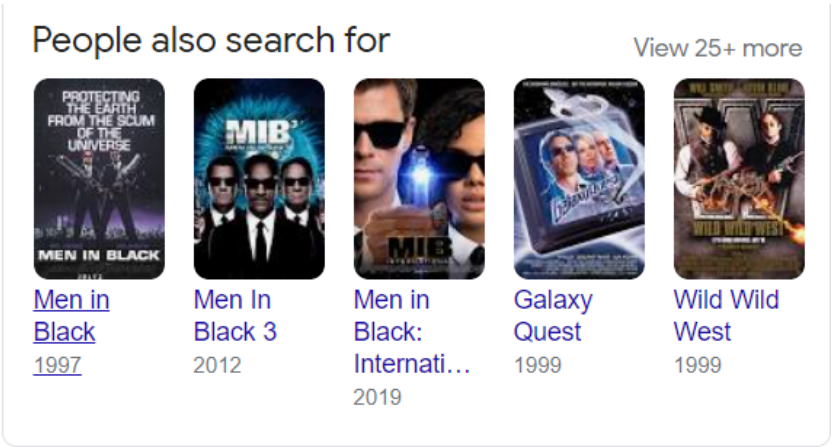

Figure 5: Original recommendations for Men in Black II

The output is accurate since Men in Black is a prequel of the target movie and share many common foci nodes such as similar name , same cast , directors , genre and country due to which it ranks the most in our program. Wild Wild West in the other hand has lesser common foci compared to 'Men in Black' due to which it ranks below it in our recommendation. Most of the recommended movies are from the country [USA] and belong to the genre [Action] , [Science fiction] which is closely related to target node. By using some user discretion the result seems correct and matches with the result available in internet. The ones not present in the original result is because those movies aren't present in our data-set.

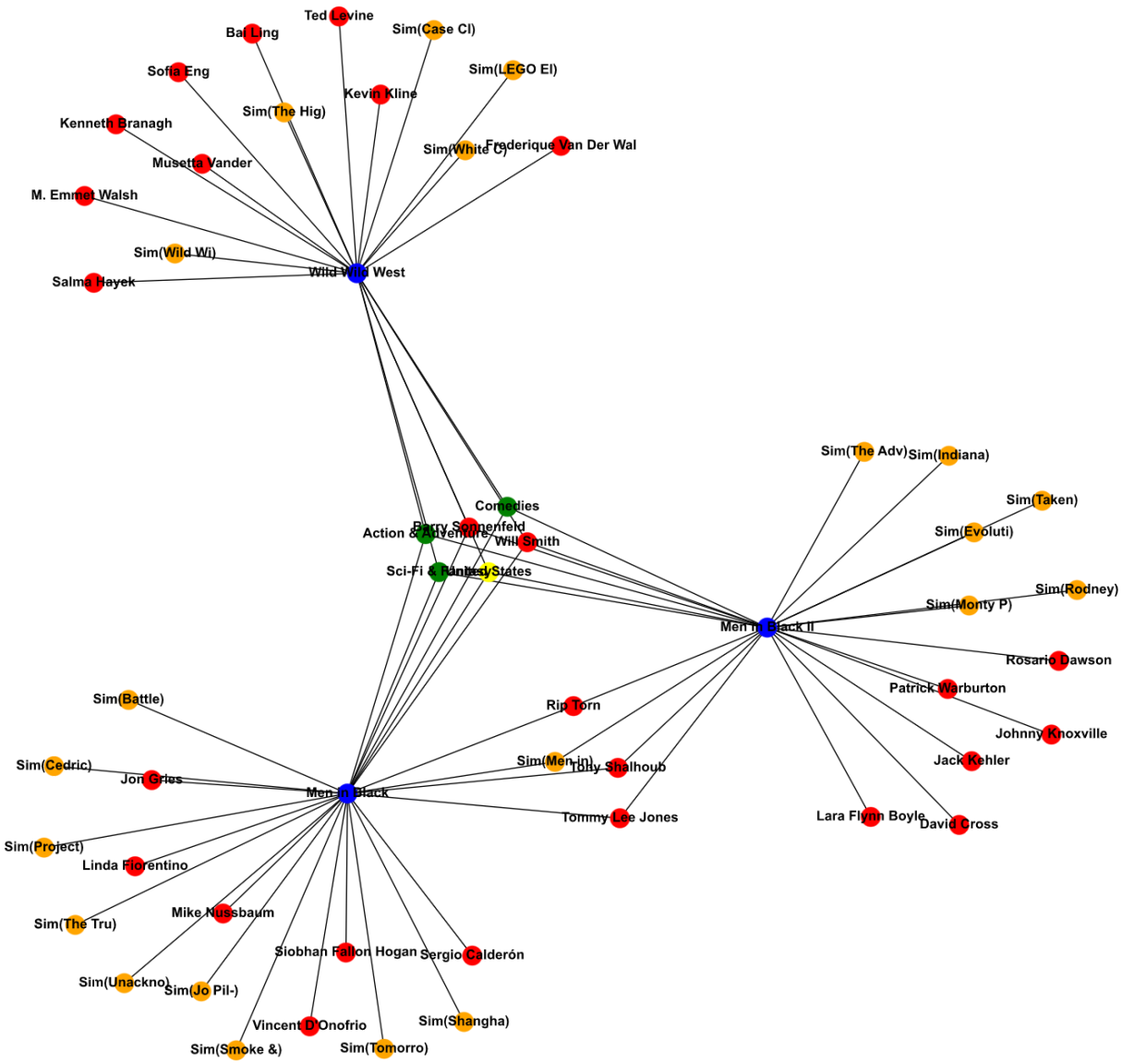

Figure 6: Neighborhood around Target Node 'Men in Black II' with the best two recommendation from the output.

Analysing the graph , we can clearly see the common foci of the first recommendation is more than second best recommendation. Hence , the adamic adar index is working perfectly with the idea that more common foci helps in more similarity value. The orange Nodes are Text similarity nodes (Eg:- sim(Men in) foci between men in black and men in black 2 is formed via text clustering). The Red Nodes are cast and directors , green for Genre , yellow for country of origin and Blue for movie node. This form of labelling if followed throughout in every subgraph figures.

```
show("Bareilly Ki Barfi")

==================================================================
 Recommendation Shows for : Bareilly Ki Barfi
==================================================================
Ek Ladki Ko Dekha Toh Aisa Laga                2.435900
Upstarts                                       2.001606
Luka Chuppi                                    1.736197
Dilwale                                        1.729006
Trip to Bhangarh: Asia's Most Haunted Place    1.723649
True Memoirs of an International Assassin      1.585269
Andhadhun                                      1.469150
In Between                                     1.442695
Fakta Ladh Mhana                               1.378134
Article 15                                     1.326576
All's Well, End's Well (2009)                  1.236653
Stree                                          1.127969
Lang Tong                                      1.094080
Made in China                                  0.993205
Queen                                          0.993205
dtype: float64
==================================================================
```

Figure 7: QUERY : Bareilly Ki Barfi

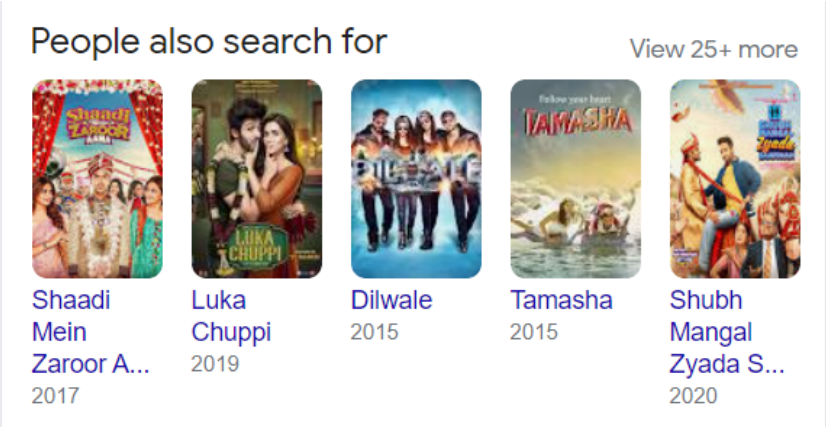

Figure 8: Original recommendations for 'Bareilly Ki Barfi'

Compared to the previous graph , this example has fewer common foci nodes in the subgraph containing the target node and the two best recommendation. 'Ek Ladki Ko dekha toh aisa laga' has 6 common foci (2 cast + 3 genre + 1 country) and 'Upstarts' has 5 common foci (1 cast + 3 genre + 1 country) with the target node. That might have given the first recommendation a slightly greater Adamic adar index over the second recommendation. The other recommendations are near perfect considering that all movies are having common country foci [INDIA] and some common cast foci such as [Kriti Sanon] , [Rajkumar Rao] and [Ayushmann Khuranna]. Moreover , most of them belong to similar genre and are mostly Comedy/Romance movies. By interpreting the result we obtained from internet we can see that 'Dilwale' and 'Luka Chuppi' came up in our output results. By user discretion this result is satisfactory.

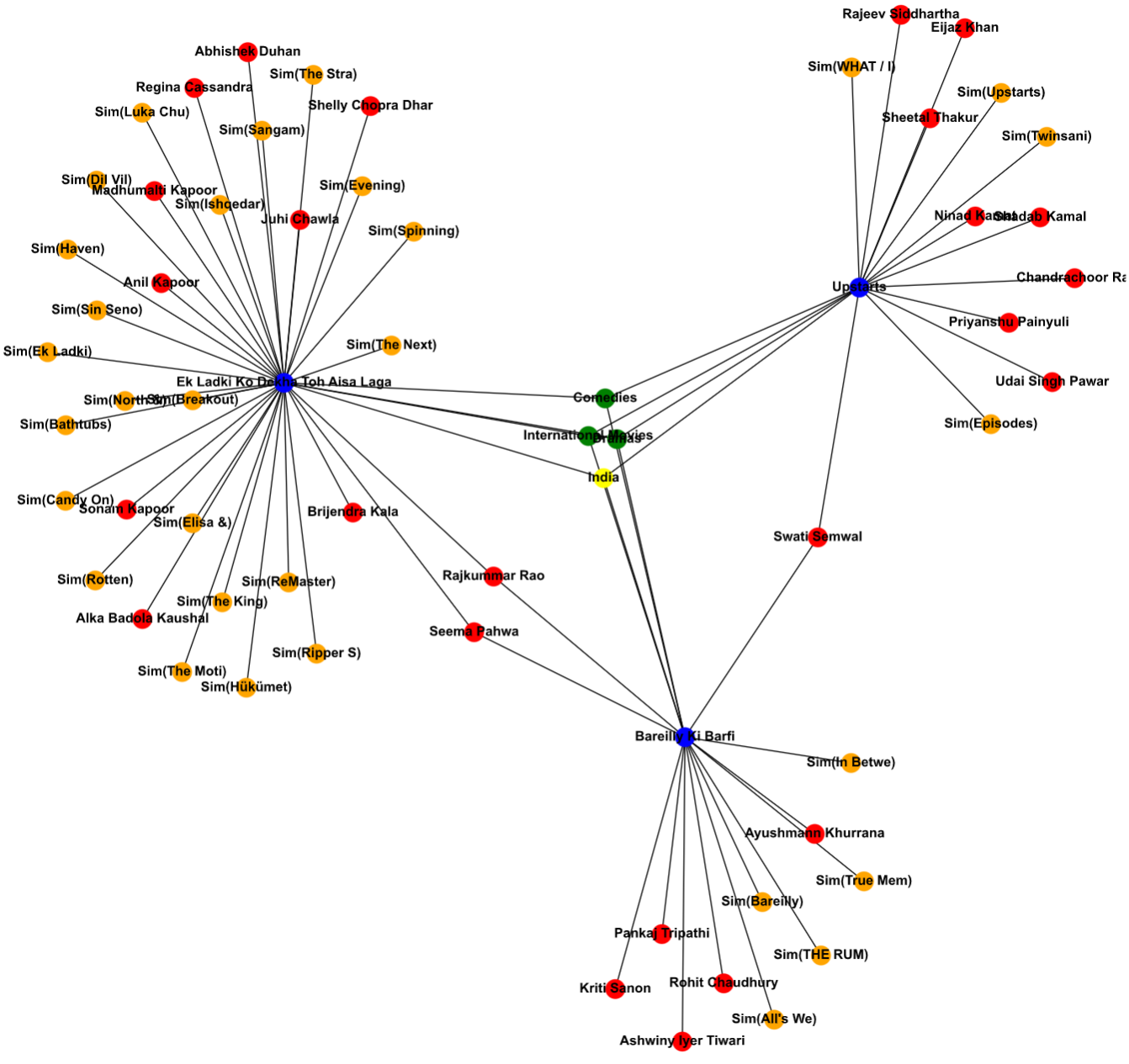

Figure 9: Neighborhood around Target Node 'Barielly ki Barfi' with the best two recommendation from the output.

In our next two sample query we used a very different type of 2 movies/shows. The first query is popular Superhero movie title 'Avengers : Infinity War' and other is 'Naruto' which is a Highly popular Japanese Anime with many sequels. We have tried to keep all 4 sample queries which vary differently so that we could do better analysis on how relevant our recommendation system is.

```
show("Avengers: Infinity War")

==================================================================
 Recommendation Shows for : Avengers: Infinity War
==================================================================
Thor: Ragnarok                                3.840104
Black Panther                                 2.105402
XXx                                           1.726493
Her                                           1.659885
Kill the Messenger                            1.569881
Kodachrome                                    1.569881
Zodiac                                        1.469869
War Horse                                     1.329432
Aliens Ate My Homework                        1.279000
IO                                            1.227442
Point Blank                                   1.194038
Wet Hot American Summer                       1.119591
Star Wars: Episode VIII: The Last Jedi        1.095150
The Talented Mr. Ripley                       1.088983
The Little Prince                             1.055629
dtype: float64
==================================================================
```

Figure 10: QUERY : Avengers Infinity War

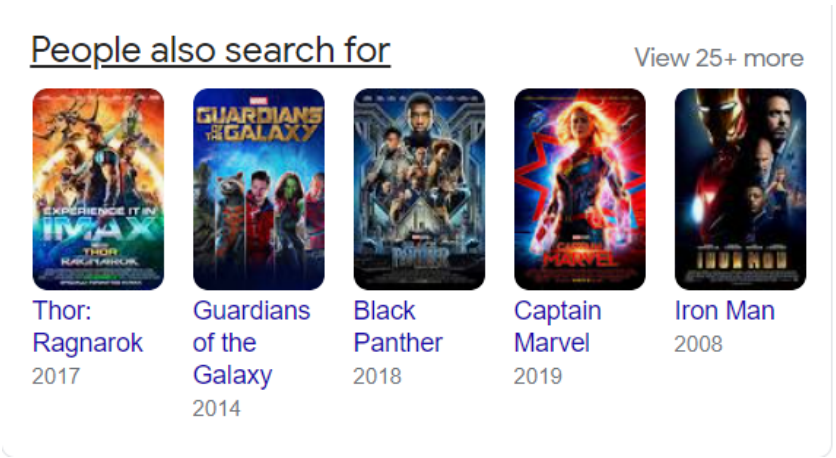

Figure 11: Original recommendations for 'Avengers : Infinity War'

The result obtained for Avengers Infinity war is near perfect considering it was able to recommend movie titles that are part of same production unit (MARVEL) even though they had great difference in title names. By analysing the Subgraph in Figure 12: its clear that there is no similarity foci for their title name (orange coloured). The recommendation have solely depended on common cast , directors and country due to the movies 'Black Panther' and 'Thor Ragnarok' being from same production unit. Moreover , The lower ranked movie in recommendation are science-fiction/ space-exploration based movies which comes in great interest of our target node. Hence , this output via user discretion is near perfect.

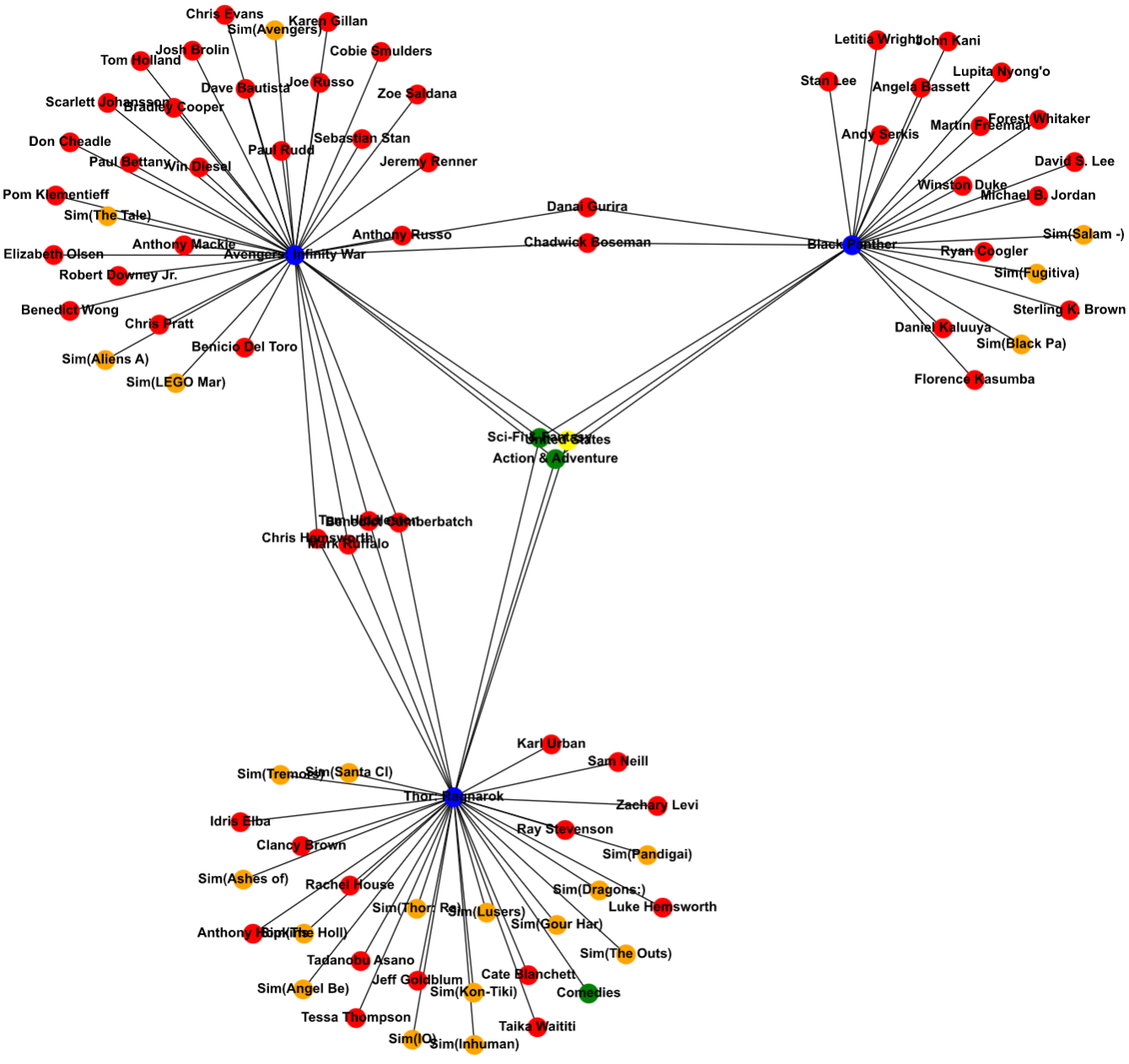

Figure 12: Neighborhood around Target Node 'Avengers : Infinity War' with the best two recommendation from the output.

```
show("Naruto")


======================================================================
 Recommendation Shows for : Naruto
======================================================================
Naruto Shippûden the Movie: Bonds                                3.910366
Naruto Shippuden : Blood Prison                                  3.585359
Naruto Shippuden: The Movie                                      3.585359
Naruto the Movie 2: Legend of the Stone of Gelel                 2.708120
Naruto the Movie 3: Guardians of the Crescent Moon Kingdom       2.675120
Naruto Shippûden the Movie: The Will of Fire                     2.575946
Saint Seiya: The Lost Canvas                                     2.399000
Fullmetal Alchemist: Brotherhood                                 1.918101
Kill la Kill                                                     1.746319
Naruto Shippuden: The Movie: The Lost Tower                      1.739102
Beyblade: Metal Fusion                                           1.532719
SWORDGAI The Animation                                           1.449345
Dino Girl Gauko                                                  1.412494
Fullmetal Alchemist                                              1.260453
Bleach                                                           1.053004
dtype: float64
======================================================================
```

Figure 13: QUERY : Naruto

Figure 14: Original recommendations for 'Naruto'

The recommendation is perfectly in sync with the original recommendation for the Anime 'Naruto'. What we have interpreted from the output and the subgraph in Figure 15 is that Adamic Adar value have improved due to extra 3-4 foci nodes which are formed due to their similar title names. Obviously the sequel and prequels tends to have nearly same prefix due to which our implementation of k-means clustering has come up really handy in the recommendation. Moreover , all other shows below in the recommendation are Japanese origin and are animated web-series which is similar in nature of our target node. By user discretion the result is very good for this query and gives as much punch as the original recommendation.

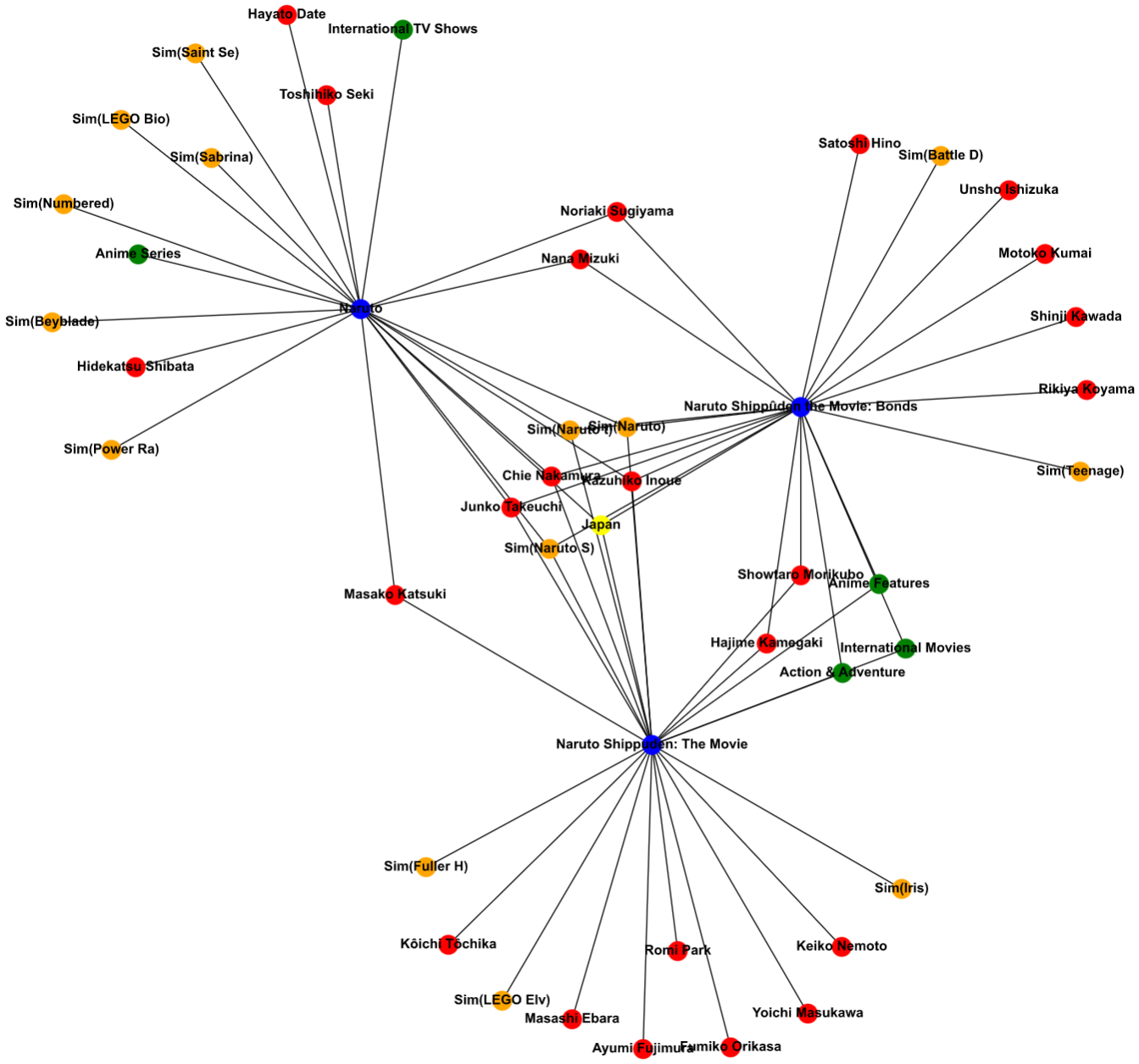

Figure 15: Neighborhood around Target Node 'Naruto' with the best two recommendation from the output.

After analysing the output graphs , we indeed realised that a small world phenomena is surely present in the graph structure with a extreme connectivity of each node to all other nodes in the graph. Some insights we obtained while doing this project is , K-means clustering stops improving the change in the adamic adar measure and become constant after some value. Here the threshold value is meant by how many k-most similar movies are to be found and added edge with the "Sim(FOCI)" when we used the text clustering process during our initial phase of implementation. We found out that nearly after k=7 , there is no significant change in the output result. If we do the same for a number k(more than 7) then the adamic adar index do not change much and only increase the density of the graph.Hence, keeping in mind the efficiency and the correctness of the result we chose the empirical value of similarity nodes to be added to be 7. Another key aspect which we understood is that the similarity measure in these type of networks come up high only for a few very closely related nodes and most of the nodes ranked below have relatively very similar adamic adar index. This can be clearly seen in all the 4 output files which we have generated. Hence , for the recommendation engine to work in a user friendly manner , we would recommend only 7 movies including the first 4 movies and randomize the recommendation of the movies from rank 5-10.This will help to ensure that the worst case recommendation is improved significantly.

3 Conclusion

Before we started this course , we were very amateur in the field of analysing graphs and understanding how critical it is in our daily life. Inspired by most other social networking sites which have great variety of random-walk algorithms for recommendation in large scale graphs,we thought of bringing up some robust approach for a relatively small scaled graph. We used the ideas that we learned from this course about homophily and existence of foci in the network in our advantage to amalgamate with Adamic Adar measure to create a recommendation engine. During this process , we learned to implement k-means clustering with TF-IDF to scrap out raw text similarities between movie names and description. This was something that was outside the domain of what was taught in our course and we were really happy to learn it on our own.Future work could focus on strategies to deal with recommending similar type of nodes that may be present in different disconnected components of the graph. A great idea of using teleportation may come in handy for such network which could be a another add-on that we may try to amalgamate into this project in future.

3.1 Team Work

All members of the team did equal work for this project. Considering the hectic schedule we had during this online semester , we did a 4 hour google meet session every week to discuss about the implementation and finding useful resources in a combined team effort.Most of the coding and writing report in latex was done in a forked document so that the load could be divided efficiently.
The only better improvement that could have happened during the whole experience making this project is that if the discussions could have been offline. Nevertheless , we tried to use the best of the available technology and resources to create this project with great determination.

References

[AA] Lada Adamic and Eytan Adar. *Adamic Adar measure*. `https://en.wikipedia.org/wiki/Adamic/Adar_index`.

[HZ14] Luyao Hou and Emma Zhong. *YouTube Video Recommendations: Analysis and Link Prediction*. Technical report. Project report for CS224. 2014.

[Sal18] Mikhail Salnikov. *K-means clustering TF-IDF*. Aug. 6, 2018. https://medium.com/@MSalnikov/text-clustering-with-k-means-and-tf-idf-f099bcf95183.

[Sha18] Parth Shah Shantanu Thakoor Anchit Gupta. *Recommending Related YouTube Videos*. Technical report. Project report for CS224. 2018.