CSC 501

Assignment 3

Subreddit Graph Data Analysis

Karan Tongay
Trishala Bhasin

Index

1. Introduction

We were being given subreddit graph data in which we had subreddits along with the hyperlinks that shows connection from source to target subreddit. We were being given 2 files namely body and head file which had hyperlinks as well as link sentiment associated with it. For our assignment we made use of body.tsv (https://snap.stanford.edu/data/soc-redditHyperlinks-body.tsv).

We carried out the preprocessing and data modelling using pandas, and then used networkX to analyze the node-links available in the data and visualize the graphs. Our choice of data model was Edge List for this assignment and our decision to choose edge list is discussed in our Data Modelling section. For further visualizing the insights related to graph data, we exported the data frames obtained to tableau. The tools used in the assignment were:

- Python 3
- NetworkX
- Tableau
- Google Colab

2. Data Modelling

2.1 Data Preprocessing

The raw data was available in the tsv format, we loaded the information in pandas and dropped the following columns:

Deleted properties column: could be better used with machine learning predictions.

Deleted rows that were dated before January 1, 2014: We decided to further segregate the information yearly in order to compare the graph connectivity trends over the year. Since the entries for 2013 were not complete, we decided to drop it as one of the parts of our stochastic filtering.

Later, we segmented the data temporally based on years from 2014-2017 to visualize the shift in node link trends over the years.

Also, since we had a huge dataset available, we decided to use body.tsv only.

We extracted the top 20 subreddits based on their in/out degrees by combining source and target. Curiosity was to know over time statistics with respect to these top 20 subreddits. Insights were meaningful and helped discover unique patterns. We have discussed more about it in the visualization section.

Also, we calculated the weights of the nodes by adding all the sentiments towards that node. This gives an idea about the most controversial and uncontroversial subreddits from the user perspective.

2.2 Data Modelling Techniques

Here, we used an exploratory approach by exploring 3 modelling techniques that are very popular in the graph data namely: Adjacency list, Adjacency matrix and edge list. We tried to model our cleaned data using all three of these techniques and compared their performances.

The adjacency list is as follows:

```
[ ] # Adjacency List
    db[db['SOURCE_SUBREDDIT'].isin(top_targets)].groupby('SOURCE_SUBREDDIT')['TARGET_SUBREDDIT'].apply(lambda x: set(x.tolist()))
```

```
SOURCE_SUBREDDIT
askreddit            {pettyrevenge, parenting, confession, cscareer...
conspiracy           {tinfoilhats, esist, russia, explainlikeimfive...
copypasta            {ireland, fivenightsatfreddys, biggerthanyouth...
drama                {intp, ireland, fivenightsatfreddys, ethereum,...
explainlikeimfive    {minecraft, morbidreality, anarchism, wallstre...
funny                {funnypics, upvoted, jokes, iama, helpmefind, ...
gaming               {rpg, tipofmyjoystick, monsterhunter, orcsmust...
iama                 {selfdrivingcars, detroitredwings, country, wi...
leagueoflegends      {minecraft, sufficiencybot, siriusgamingleague...
legaladvice          {miata, pettyrevenge, ireland, tagpro, parenti...
mhoc                 {mhocstrangersbar, mwnn, mhocmeta, mhocsatire,...
news                 {worldnews, bestof, syriancivilwar, pics, poli...
outoftheloop         {ireland, parenting, nexus6p, jellyfish, bindi...
pics                                {fox, horror, changelog, modsupport}
subredditdrama       {pettyrevenge, ireland, fivenightsatfreddys, t...
subredditoftheday    {phish, gifsofotters, churning, bigbangcomics,...
videos               {confirmedtestcss, politicalvideo, videos_disc...
worldnews            {europe, iama, ukraine, iraqconflict, euromaid...
writingprompts       {androidapps, explainlikeimfive, screenwriting...
Name: TARGET_SUBREDDIT, dtype: object
```

Fig 1. Adjacency List

We learned that adjacency list, being a hybrid of adjacency matrix and edge list would have been a good approach theoretically, as it is faster, in terms of finding adjacent edges. But since our data that we picked for analysis is smaller, we made use of edge list.

We also implemented adjacency matrix, but due to its high space complexity which is O(n^2), which in our case is (35776 X 35776) (35776 is the unique number of nodes in the entire dataset) and the nature of the matrix obtained was sparse, it was not an ideal choice for data model. The snap from our adjacency matrix is as follows:

[ ]

	askreddit	iama	subredditdrama	writingprompts	outoftheloop	pics	videos	today
askreddit	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
iama	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
subredditdrama	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
writingprompts	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
outoftheloop	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
pics	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
videos	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
todayilearned	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
funny	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
gaming	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
leagueoflegends	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
copypasta	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
conspiracy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
worldnews	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
drama	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
explainlikeimfive	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
mhoc	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Fig 2. Adjacency Matrix

Our Approach: Edge List

Since we planned on conducting a majority of our analysis on a subset of data by finding out top 20 hotspot subreddits (the subreddits that have a maximum incoming and outgoing nodes), we decided to choose edge list. Our idea of using edge list was an attempt to associate our approach with the modern practices and to deal with its challenges using the indexing techniques mentioned in the LinkedIn research paper.

3. Algorithmic Considerations

Performance of Adjacency Matrix in our case:

- Adjacency matrix uses O(n^2) memory
- It is fast to lookup and check for presence or absence of a specific edge
- between any two nodes O(1)
- Too slow to iterate over the entire set of edges
- Time complexity for adding new edge is O(1)
- The entire dataset took approximately 5 GB of space to store the adjacency matrix, mostly because it was a sparse matrix and was way too sparse for the top 20 subreddits.

Performance of Adjacency List in our case:

- Memory usage depends on the number of edges (not number of nodes like in adjacency matrix),
- Saved a lot of memory since the adjacency matrix was sparse.
- It is fast to iterate over all edges as we can directly access any node neighbors.
- Time complexity for adding new edge is O(1)
- Space complexity is O(n+m) [where n is no. of vertices and m is no. of edges].

Performance of Edge List in our case:

- In our case, edge list proved to be faster than the above two data models.
- The time complexity is O(1) for accessing vertices of a particular edge.
- Since we followed edge based graph model, edge list was our preferred choice of data model.
- Space complexity is O(n+m) [where n is no. of vertices and m is no. of edges]
- Limitation: Although we realized that finding adjacent edges can lead to traversing entire edge list.

4. Visualization

For the visual analysis of our graph data, we used 2 main tools namely the network X for leveraging the graph style visualization ability of python and tableau, that we used to show some statistical insights that we received from data analysis that could potentially be used in dashboard style analysis of the graph data.

The network X visualization and our data processing link is as follows:
Google Colab Notebook

The tableau visualization link is as follows :
Tableau Visualization

4.1 Visualization of data as a graph:

Since graph data works on the concept of connectivity, We decided to look at top 20 subreddit threads and analysed their connectivity across the entire dataset and how its connectivity trends changed between the span of 2014 to 2017. This aligns with the ***temporal aspect*** of data analysis which we learnt in the previous assignment. We analysed that information with the help of network x. Later, we extracted the graph data being obtained and analyzed it for understanding the sentiment associated with each post as it appears in various subreddit communities. The visualization for the same is as follows:

(Due to high density of nodes, the graph obtained from Network X is not clear in the image but we have shared the link to these visualizations).

Our visual encodings for the following visualizations are:

1. The size of the green nodes represents the weight of the subreddit.

2. The orange nodes represents the popular source subreddits that are highly connected with the top 20 target subreddits.

3. The gray nodes represent the other source subreddits which are not interacting with other top 20 target subreddits.

4.1.1 Visualization 1 (2014):

The interesting node to observe here is “mhoc”. We can see that “mhoc” is one of the closely connected nodes to the Top 20 community in 2014.

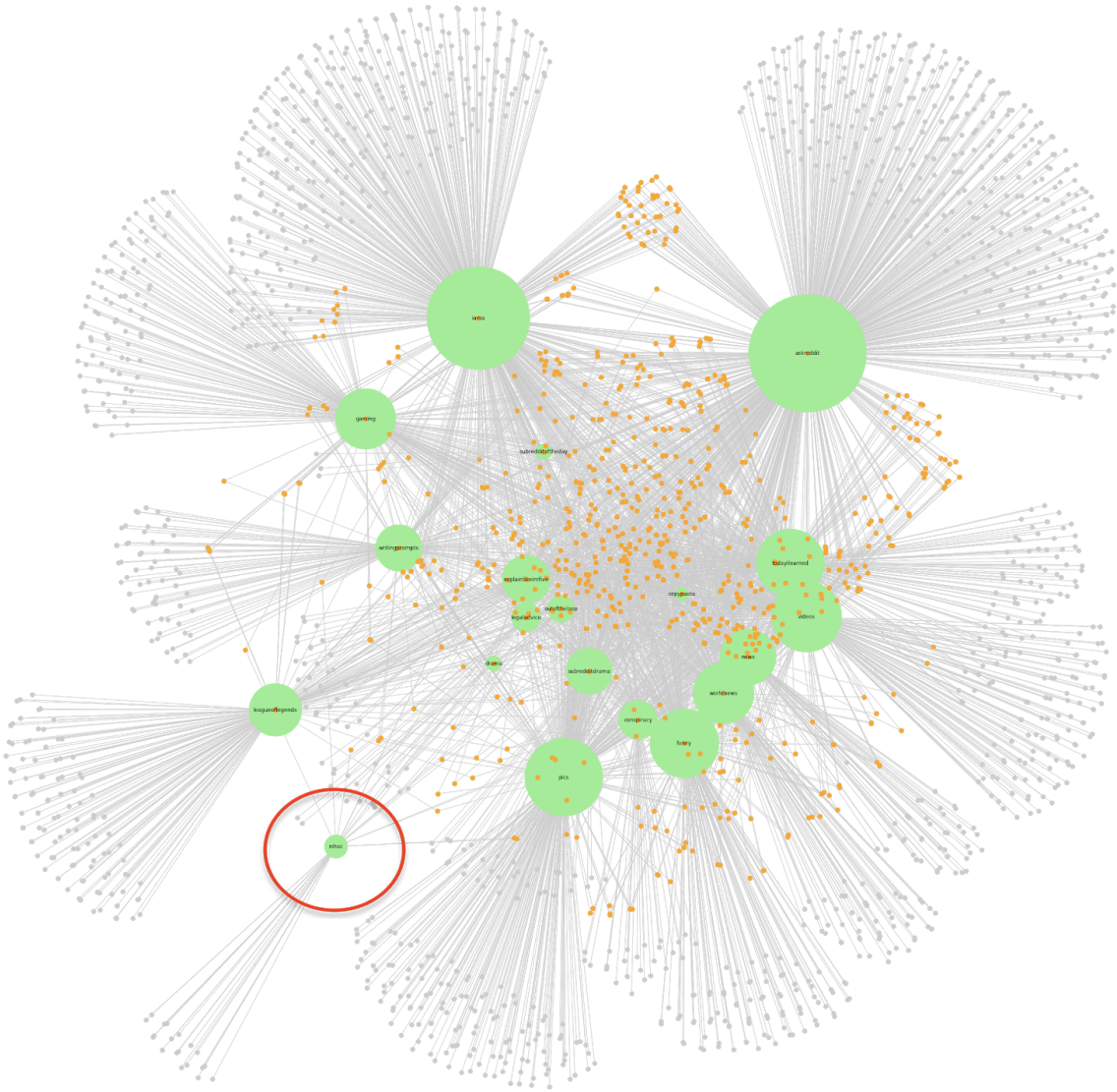

Fig. 3 Top 20 Subreddit Distribution in 2014

4.1.2 Visualization (2015):

In 2015, we can see that “mhoc” is slightly moving away from the top 20 community. There are a very few posts targeted towards “mhoc”.

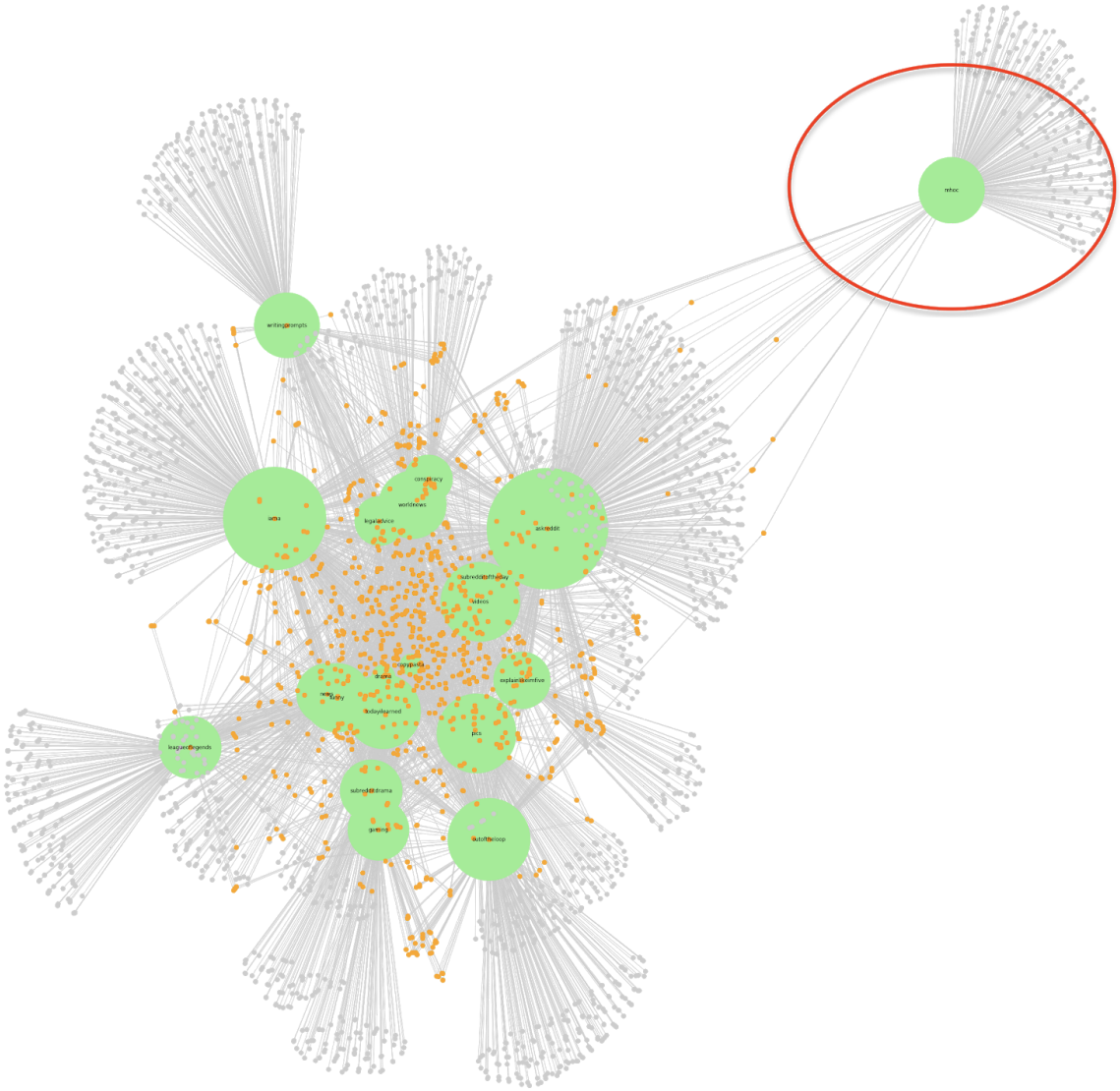

Fig. 4 Top 20 Subreddit Distribution in 2015

4.1.3 Visualization (2016):

In 2016, we see that “mhoc” still survives to be a part of Top 20 closely connected subreddit communities.

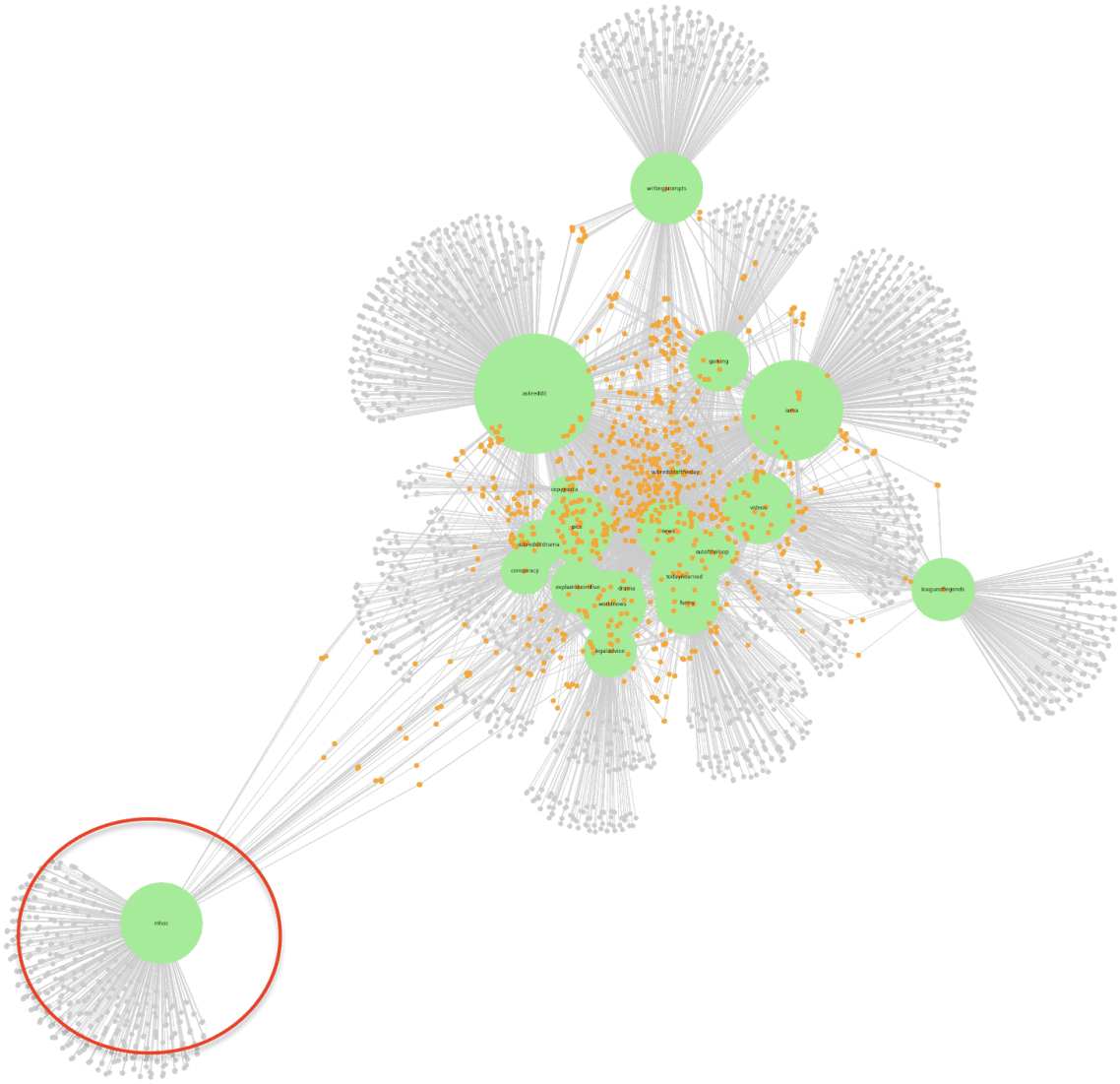

Fig. 5 Top 20 Subreddit Distribution in 2016

4.1.4 Visualization (2017):

In 2017, we see that “mhoc” drifted away from the top 20 subreddit community.

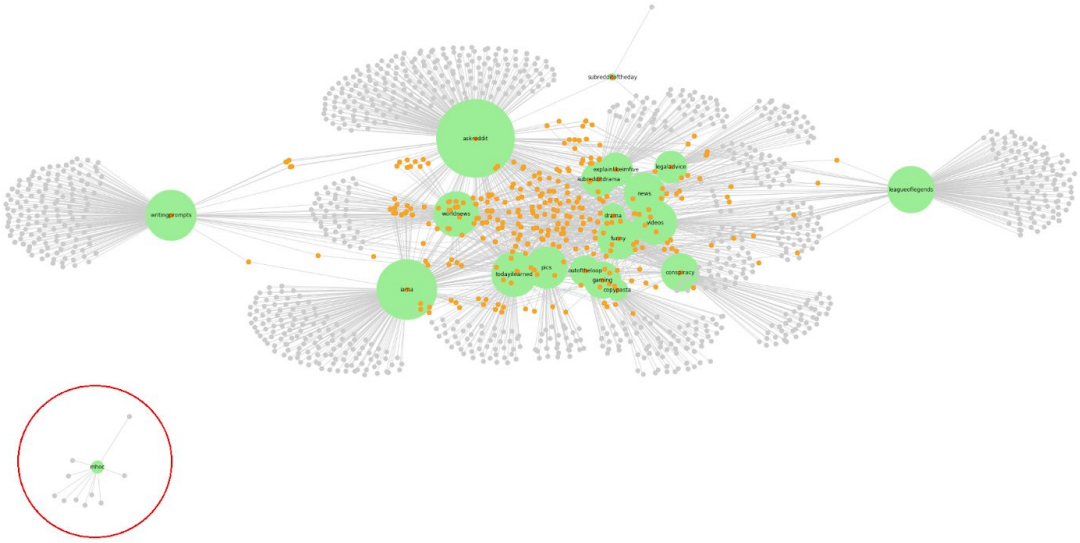

Fig. 6 Top 20 Subreddit Distribution in 2017

Similarly, we can use this style of visualization in order to observe the trends existing in the subreddit graph community.

4.2 Dashboard style visualization:

We decided to use this style of visualization in order to show trends existing in our data as it is a familiar format for visualization process and it simplifies the insights which appear complex if being done using graph clustering. The visualizations here are being done by exporting our data frames that we obtained from network X to tableau. It gives us a summary of demographics of the data better. The link to our tableau dashboard is as follows:
[Tableau Visualization]

Tableau Visualization Insights:

4.2.1 Visualization 1 (Sentiment Distribution for top 20 subreddits):

Here, for our most popular top 20 subreddits, we extracted the sentiments associated with it. It helped us identify with what sentiment was a given post perceived as it was reposted from the source subreddit to target subreddit. For instance, here, we can clearly see that the askreddit community has received a total of 7,329 posts out of which 1,784 had a negative sentiment and 5545 had positive sentiment. This information can be used to identify general attitude of the popular threads towards the posts and can be used greatly while doing sentiment analysis of a given network of community.

4.2.2 Visualization 2 (Number of triangles the top 20 subreddits are associated with):

Here, we first took our top 20 most engaged subreddits and calculated their existence in a triangle from the entire dataset. The more it exist in a triangle, the more engaging the subreddit is. This information can be useful when calculating associations between various subreddits with respect to the movement of post in a given graphical data.

4.2.3 Visualization 3 (Heat map of our top 20 target subreddits based on count of incoming posts):

We created a heat map that uses size and color as a visual encoding to determine the targets which have the greatest number of incoming subreddits from the source. This information can be useful in understanding the numerical composition of the graph data and the nodes that contribute the most in the subreddits community.

5. Relationship to the Graph Data Challenges

Graph data visualization is a major challenge as mentioned by T. von Landesberger et.al. in “Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges” research paper. One of the challenges we faced was representing our entire graph model in a layout. The visualization can be less insightful if there are thousands of nodes involved in a graph data model

and we have a limited space for laying out the graph. Therefore, we took an approach to perform graph filtering. The paper mentioned two approaches for performing graph filtering. For our use case, we decided to go with Stochastic

Graph Filtering approach which is based on random selection of nodes and edges from the original graph. Keeping our scope focussed on the top 20 subreddits in the entire data, we decided to use this subset of top 20 subreddits which highlights our stochastic filtering approach. Although, we realized it was still a challenge to visualize even this subset of data on the limited space. The possible solution to this would have been graph aggregation, but we took this requirement of analyzing the behaviour of these top 20 subreddits as a challenge for ourselves.

Below are the challenges we faced which can be related to the graph data challenges in general:

- **Human Perception:**
  Another challenge is the perception of the graph visualization, it strongly depends on the human perception capabilities. Additionally, choosing an appropriate type of graph visualization can also be challenging given that there are many different types of graph visualization techniques growing substantially.

- **Scalability problems:**
  As the number of nodes increases in the data model, its visualization and layout computing can become expensive.

- **Displaying nodes with labels:**
  While visualizing graph data, it would also be insightful to show the data associated with each node maybe in the form of labels. The challenge is, even while visualizing smaller graphs if we try to add the labels to the nodes, the graph visualization can lead to cluttered visuals.