University of Victoria

CSC 501

Report (Assignment - 3)

Submitted by:

Name	**Student ID**
SIDDHARTH CHADDA	V00947906
DHRUVRAJ SINGH	V00970352
ABHISHEK KATHURIA	V00959831

Introduction

Reddit is an American online community of message forums that is organized into over one million user-created and user-moderated communities known as subreddits. Alongside mainstream subreddits for discussing scientific discoveries and internet memes, during the recent 2016 US elations Reddit has seen an increase in 'Toxic and Harmful' subreddits, that promote violence, hate speech and racism in communities.

The goal of our investigation is to come up with appropriate data modeling, data storage and data visualization techniques to help us analyze the large SNAP Stanford reddit dataset in an efficient and scalable way, and identify these harmful and toxic subreddits.

Dataset Background

To facilitate our investigation, we are using the SNAP Sandford 'Social Network: Reddit Hyperlink Network' dataset. This dataset is a collection of monthly user interaction networks derived from multiple publicly available subreddit communities on reddit.com from 2014 to 2017. The datset consists of two TSV files, each of which provides links from the source sub-reddit to the target sub-reddit. Both files include hyperlinks that are referenced in the body and title of a subreddit.

Conceptual Data Modelling

From our investigation we wish identify the toxic and harmful subreddit groups and to do this we first need to understand the underlying realtionship between 2016 US elections atmosphere the rise in hate speech and toxicity on the internet. We will also try and investigate how some political actors might have promoted or exploited this rise in online toxicity to their benefit.

Hence given the nature of our investigation we believe Network Graph based Data models like Adjacency Matrix, Adjacency List and Edge List are the most suitable data models.

Reason for Using Network Graph based Data Modeling:
In order to identify the Entities and Relationships in our investigation, we need to find answers to questions like "How do the various subreddits groups interact with other subreddit groups on Reddit?", "Which are the most influential/most active subreddit groups in the social network?", "What is the underlying relationship between different subreddit groups?", "What cluster/family/ topic does the subreddit belong to? Examples of topics can be Donald Trump, Race, Guns etc.", "What is the polarity of user sentiment (positive or negative) of the subreddit groups?". Such type of questions will be better answer using a network graph-based data modeling technique.

1. A network graph-based data model provides a bird's eye view of interactions between subreddit groups and allow us to make sense of interactions between different subreddits groups

2. Help us identify the most Active/ Influential subreddit groups in the social graph.

3. Help us gauge the level/ type of sentiment towards a particular subreddit group.

4. Help us identify the Clusters / Family group that particular subreddit belongs to, thereby helping us out to identify more toxic and harmful subreddit groups belonging to the same topic cluster.

Basic Anatomy of our Network Graph based Conceptual Data Models:

Nodes / Vertices:

Represented by circles in our Network Graph. A node is an Entity in a graph. In our investigation each node/ vertex will represent a single subreddit group.

Edges:

Represented by lines connecting the nodes/ vertices in our graph. We are using Undirected graphs in our case. In an undirected graph the edge represents a 2-way connection or Relationship between 2 nodes / subreddit groups. In our case the edge represents the polarity of the sentiment of the source community subreddit post towards the target community subreddit. The edges of our graph data model are weighted based on the values of the "LINK_SENTIMENT" attribute, with values +1 and -1. A value of +1 represents a positive sentiment of the post whereas the value of -1 represents a negative sentiment post.

Layout:

The layout of this graph has to do with the way the nodes are distributed. For example, tightly connected nodes tend to be closer together. In order to visualize our graphical data models in the best possible way we will experiment with different kinds of layout models ranging from Circular layout to Tree based connected layout to a Grid based layout.

Node Size:

The node size is determined by the sum of responses and comments by the respective subreddit group. We are taking the sum of both incoming and outgoing responses, since we just want to check the activity/ user engagement level of the node/subreddit group. Larger size indicates the subreddit is more Active/ Influential or produces more responses, a Smaller size indicates the subreddit is less active.

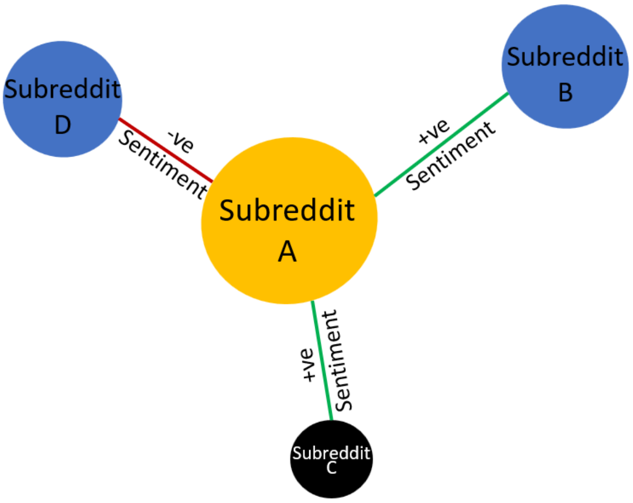
Node Colour:

Different colours represent different level of Activity/ Influence of the nodes/ subreddits groups. The three levels of activity are namely, Very Active: Yellow, Moderately Active: Blue, Less Active: Black.

Edge Colour:

Edge colour indicates of polarity of the sentiment of the source community subreddit post towards the target community subreddit. Red colour meaning a Negative (-) sentiment and a green colour meaning a Positive (+) sentiment.

The below image visualizes the proposed network graph conceptual model in our investigation:



Logical Data Modelling

For building the logical model of our investigation we will implement the below graph-based data models:

1. Edge List
2. Adjacency List
3. Adjacency Matrix

Modelling the Edge List:

We will first begin by building a simple Edge List model and then fine tune it to our needs. An edge list is a list-based graph data structure that contains the list of all the edges in the graph model.

From the above definition, a simple Edge List model can be thought of as list of edges in the form:

[E1, E2, E3………, En]

Here 'E', represents the Edges in the graph model

On further analysis, we can draw the edges as a connection between Source node and Target node, in the form:

[(V1, V2), (V2, V3), (V3, V4) ……………………….. (Vn-1, Vn)]

Here 'V', represents the Vertices in the graph model.

Modifying the Edge List data model to compensate for Weighted Edges:

The edges in our case are weighted with weights of +1 (positive sentiment) and -1 (negative sentiment). Hence, we must also take into account the ***Weight*** of the edges, thus in our case the edge list will be modeled as:

[(V1, V2, W1), (V2, V3, W2), (V3, V4, W4) …………………… (Vn-1, Vn,W4)]

Here 'W', represents the Weights of the Edges in the graph model.

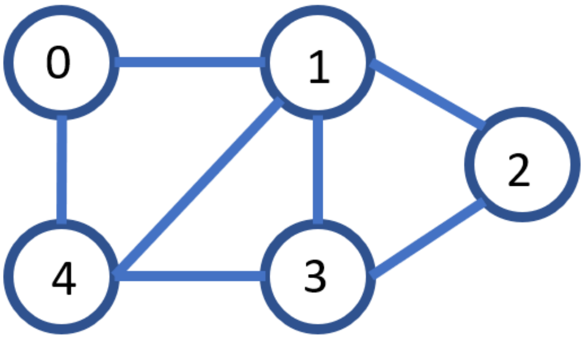
Modelling the Adjacency Matrix:

Similar to above, we will first build a simple Adjacency Matrix model and fine tune it to our needs. We can think of Adjacency Matrix as a 2D array matrix of size |V x V| use to represent a network graph data model, here V is the number of vertices in the graph.

For our investigation we are using undirected graphs, hence in our case an Adjacency Matrix if there exist a relationship between 2 nodes/ vertices V1 and V2, the positions [V1, V2] and [V2, V1] both will contain a value of 1 in the square matrix and the remaining positions where there is no edges between the nodes they will be filled by 0s.

This modelling technique can be explained though the below illustration:

For the below undirected network graph:



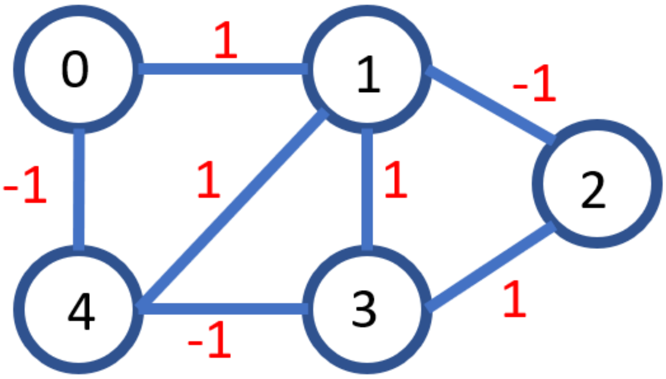
The Adjacency Matrix will look like:

	0	**1**	**2**	**3**	**4**
0	0	1	0	0	1
1	1	0	1	1	1
2	0	1	0	1	0
3	0	1	1	0	1
4	1	1	0	1	0

Modifying the Adjacency Matrix data model to compensate for Weighted Edges:

Since our undirected graph has ***weighted*** edges hence, in our case the valid edge value positions in the adjacency matrix will be replaced by their weights.

Thus, for the below ***weighted*** undirected network graph:



The Adjacency Matrix in our case will look like:

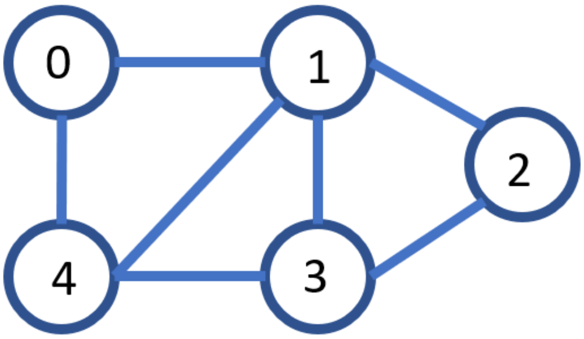
	0	**1**	**2**	**3**	**4**
0	0	1	0	0	-1
1	1	0	-1	1	1
2	0	-1	0	1	0
3	0	1	1	0	-1
4	-1	1	0	-1	0

<u>Modelling the Adjacency List:</u>

Similar to above, we will first start by defining a simple Adjacency List model and modify it to our needs. Adjacency List can be thought of as an array of Linked Lists, where the size of the array is equal to the number of vertices/nodes in the graph.

The index of the array represents the Nodes/Vertices of the graph and the elements in the linked list at that particular index represents the list of nodes which form an edge with the vertex at the index.

Going by this simple definition we can create a simplistic model of an adjacency list of the above unweighted graph as:



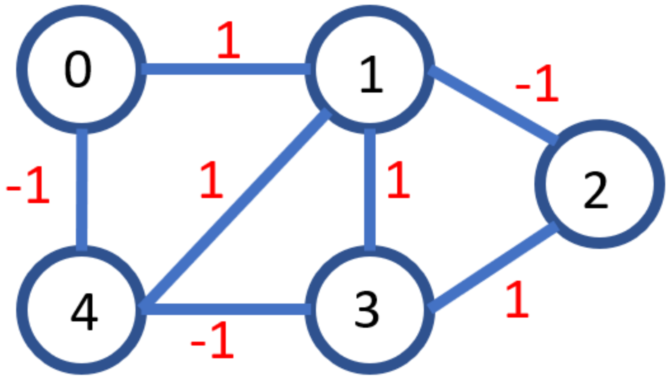
The Adjacency List will look like:

0	---->	1		---->	4	/								
1	---->	0		---->	4		---->	2		---->	3	/		
2	---->	1		---->	3	/								
3	---->	1		---->	4		---->	2	/					
4	---->	3		---->	0		---->	1	/					

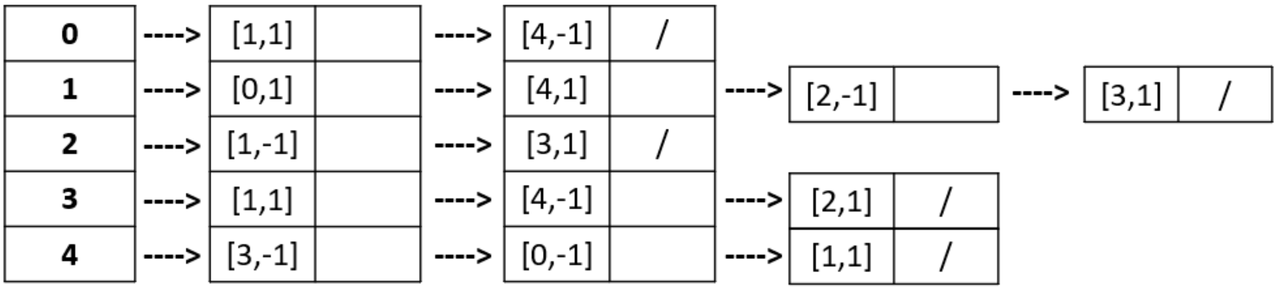
Modifying the Adjacency List data model to compensate for Weighted Edges:

We can modify the Adjacency List to include the edge weights by incorporating the weights inside the cells as nested list object.

Thus, for the below ***weighted*** undirected network graph:



The Adjacency List in our case will look like:



Experiments on Scalability

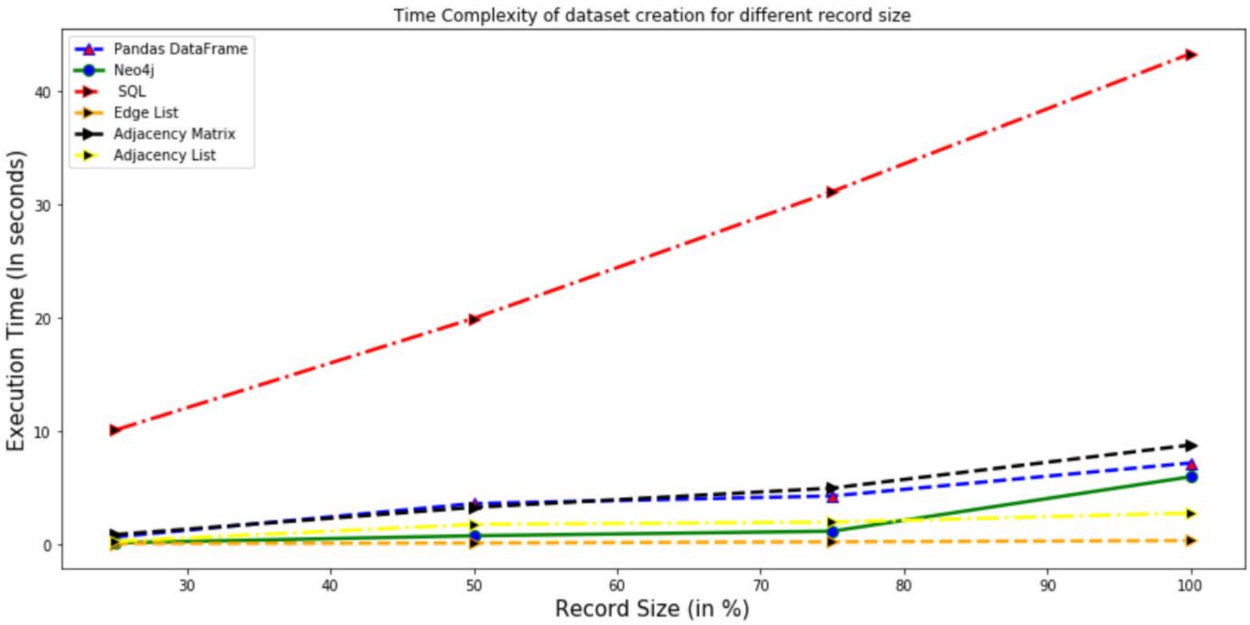
In this section we will compare the perfomance of our 3 graphical data structures against each other on the basis of data insertion time, data fetch time, and storage space optimization.

We will also compare the performance of our 3 graph data models against conventional data structures like SQL tables, Dataframe and Neo4j and we will observe the advantages of using graph data structures for analyzing graph data instead of conventional data structures.

Through the above activities we will be able to justify our conceptual model if using these 3 graph data structures and more closely explore their scalability.

Experiment 1: Comparing the Execution Time for **Insertion of records** of different sizes using the following:

- Graph Data Structures - Edge List, Adjacency Matrix, Adjacency List
- Graphical tool - Neo4j (High-performance graph database tool)
- Conventional Data Structures - SQL Queries and Pandas based DataFrame



Graph. 1 Time Complexity of Insertion for different record size

Insights:

1. It can be clearly observed from the below graph that as the record size of the dataset increases, the execution time for creation of records in the table using SQL queries (indicated in red) increases steeply as compared to the execution times for other data structures. Considering just the 25% of the total records, it is evident that the time for insertion using Pandas DataFrame is 0.7 seconds, for Edge List it is 0.11 seconds, for Adjacency list it is 0.3, for adjacency matrix it is 0.9 seconds whereas for SQL queries, it is 36.5 seconds which is way higher for insertion of small chunk of records in the table.

2. We can also observe that the insertion time using Edge List and Adjacency list remains nearly constant for the different number of records which compliments its optimality for representation of graph data.
3. It is also noticeable from the figure that when all the records were inserted, the time taken by Edge List and Adjacency List was just 0.3 and 2.8 seconds but for the Adjacency matrix and Neo4j, it is 8.8 and 6 seconds respectively.

Conclusion

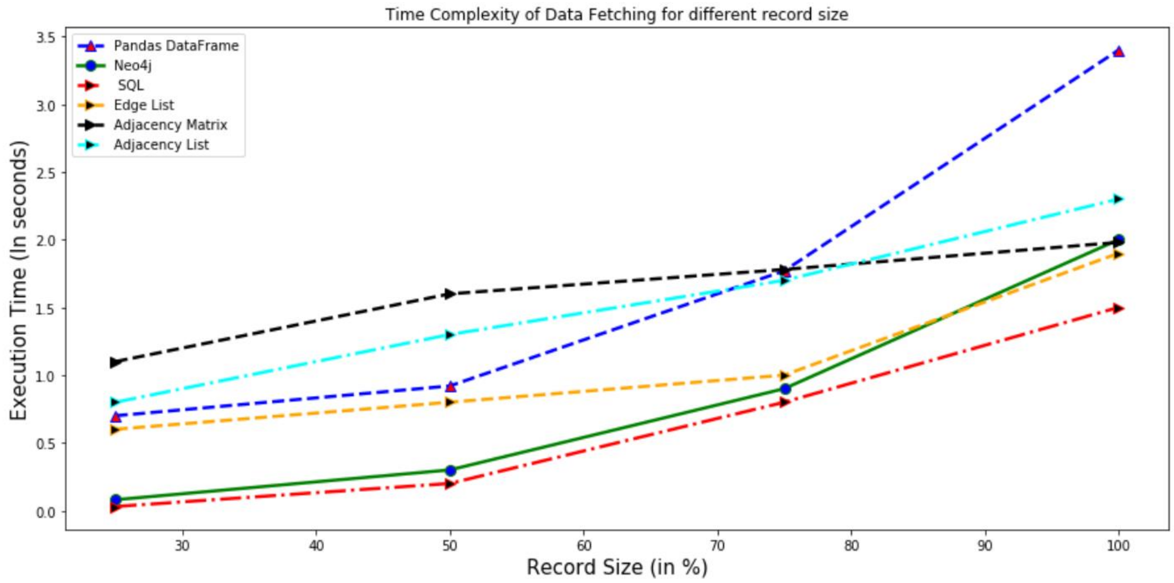
This shows that for creation and representation of records, Edge Lists are most optimized followed by Adjacency Lists.

Summarizing the Execution Time for Insertion (In Seconds)

% of Records inserted	Execution time using Pandas Dataframe	Execution time using SQL queries	Execution time using Neo4j	Execution time using Edge List	Execution time using Adjacency Matrix	Execution time using Adjacency List
25%	0.7	10.09	0.17	0.11	0.9	0.3
50%	3.6	19.9	0.8	0.16	3.3	1.8
75%	4.3	31.1	1.2	0.26	5	2
100%	7.21	43.2	6	0.37	8.8	2.8

Experiment 2: Comparing the Execution Time for **Fetching of records** of different sizes using the following:

- Graph Data Structures - Edge List, Adjacency Matrix, Adjacency List
- Graphical tool - Neo4j
- Conventional Data Structures - SQL Queries and Pandas based DataFrame



Graph. 2 Time Complexity of Sequential Fetching for different record size

Insights:

1. It can be clearly observed from the below graph that as the record size of the dataset increases, the execution time for fetching of records in the table using Pandas DataFrame (indicated in blue) increases steeply as compared to the execution times for other data structures. Considering just the 25% of the total records, it is evident that the time for fetching using SQL queries is 0.0.03 seconds, for Edge List it is 0.6 seconds, for Adjacency list it is 0.8, for adjacency matrix it is 1.1 seconds.

2. We can also observe that the fetch time using SQL, Neo4j and Edge Lists is the least and is comparable to each other. Hence, this compliments the choice of our data model to utilize Edge List as a Data Structure for representation of graph data.

3. It is also noticeable from the figure that when all the records were inserted, the time taken by SQL, Edge List, Adjacency Matrix was just 1.5, 1.29 and 1.98 seconds but for the Adjacency List and Neo4j, it is 2.3 and 2.0 seconds respectively.

Conclusion:

This shows that for fetching the records sequentially, Edge Lists better than the other graph data structures like Adjacency List and Adjacency Matrix and are extremely optimized as they are able to handle large volumetric graphical data with ease.

Summarizing the Execution Time for Fetching (In Seconds)

% of Records inserted	Execution time using Pandas Dataframe	Execution time using SQL queries	Execution time using Neo4j	Execution time using Edge List	Execution time using Adjacency Matrix	Execution time using Adjacency List
25%	0.7	0.03	0.08	0.6	1.1	0.8
50%	0.92	0.2	0.3	0.8	1.6	1.3
75%	1.77	0.8	0.9	1	1.78	1.7

100%	3.4	1.5	2	1.29	1.98	2.3

Experiment 3: Calculating the memory size of the Graphical data

Insights:

1. In the below table, we show the comparison between the memory size of the Graphical data. The memory size for adjacency list was the least with 1.1 MB whereas it was highest for adjacency matrix with 1169 MB.
2. We can clearly note that adjacency matrix and edge list (1.2 MB) have nearly identical space complexity. It also shows that they are performing better in all the prospects such as space and time complexity.

Memory Size of Graphical Data

Type of Database	Memory size
Pandas DataFrame	382.3 MB
SQLite3 database	12.27 MB
Edge List	1.2 MB
Adjacency matrix	1169.42 MB
Adjacency List	1.1 MB

Conclusion:

From the above we can see that our proposed graph data structures like Edge List and Adjacency List are extremely optimized for storage and only take a fraction of storage space when compared to conventional data structures like SQL tables or Dataframes.

Overall Conclusion:

1. *From the above 3 Experiments we can clearly see that our proposed Graph based data structures like Edge List and Adjacency List are extremely optimized for our current graphical data problem. Data Insertion and data fetching is done very fast using Edge List or Adjacency List and these data model are also extremely storage efficient.*

2. *Out of 3 proposed graph data models we can see that Edge List gives us the best Perfomance for our problem case.*

Justification for Choosing Graph Data structures for Data Modelling with regards to Scalability

- **Conceptual Model Approach**: *Represent and handle graph data using graphical data structures such as edge list or adjacency list.*

According to the given Social Media Graph Data Analysis problem, conceptually thinking the conventional Data Structures will not be able to support our investigation in a scalable way as this requires storing complex representations of the graph data. Hence for conventional Data structures would be required to continuously call nested queries to store the data. This would in turn require huge amount of time and resources.

Our conceptual modelling approach considered using Edge lists, Adjacency Matrix and Adjacency List for representing graphical data. As a further experiment, we also used Neo4j (which is a renowned tool for representing and visualizing graphical data) to add comparisons.

- **Result**: *Experimental Scalability supports our conceptual model hypothesis*

As shown above Graphical Data Structures are extremely scalable and capable of handling large data sets with ease and they even outperform conventional data structures in every experiment and even challenge the industry appreciated graph data visualization tool Neo4j.

Justification for Choosing Graph Data structures for Data Analysis Investigation with regards to scalability

- **Investigation Approach**: *Represent and visualize the relationships in the graph data using graphical data structures such as edge list or adjacency list.*

Here, we took our hypothesis that for representing and establishing relationships in our graphical data such as identifying subreddits, identifying linkage between subreddit groups, and others, would be better by using Graphical data structures like edge lists or adjacency lists.

  - ***Result:*** *Experimental Scalability supports our investigation hypothesis*

We can clearly see from experiments 2 and 3 that the scalability of our model supports the Adjacency lists and edge lists. This is because these two graphical data structures have the minimum data fetch time and occupy the least memory space.

Hence, it can be justified that for carrying our investigations efficiently, we can make use of this data structures over our conventional data structures such as Panda DataFrame or SQL. This is because for creating complex visual relationships graphs, we would have to continuously remodel our database again and again. Therefore, Edge lists and Adjacency list provide the fastest and the most space effective route for investigations even surpassing adjacency matrix which although provide a descent fetching time, but are not space efficient (as shown in experiment 3).

Exploratory Data Analysis

Plot 1: Relationship between active subreddits and Donald Trump during 2016 US elections.

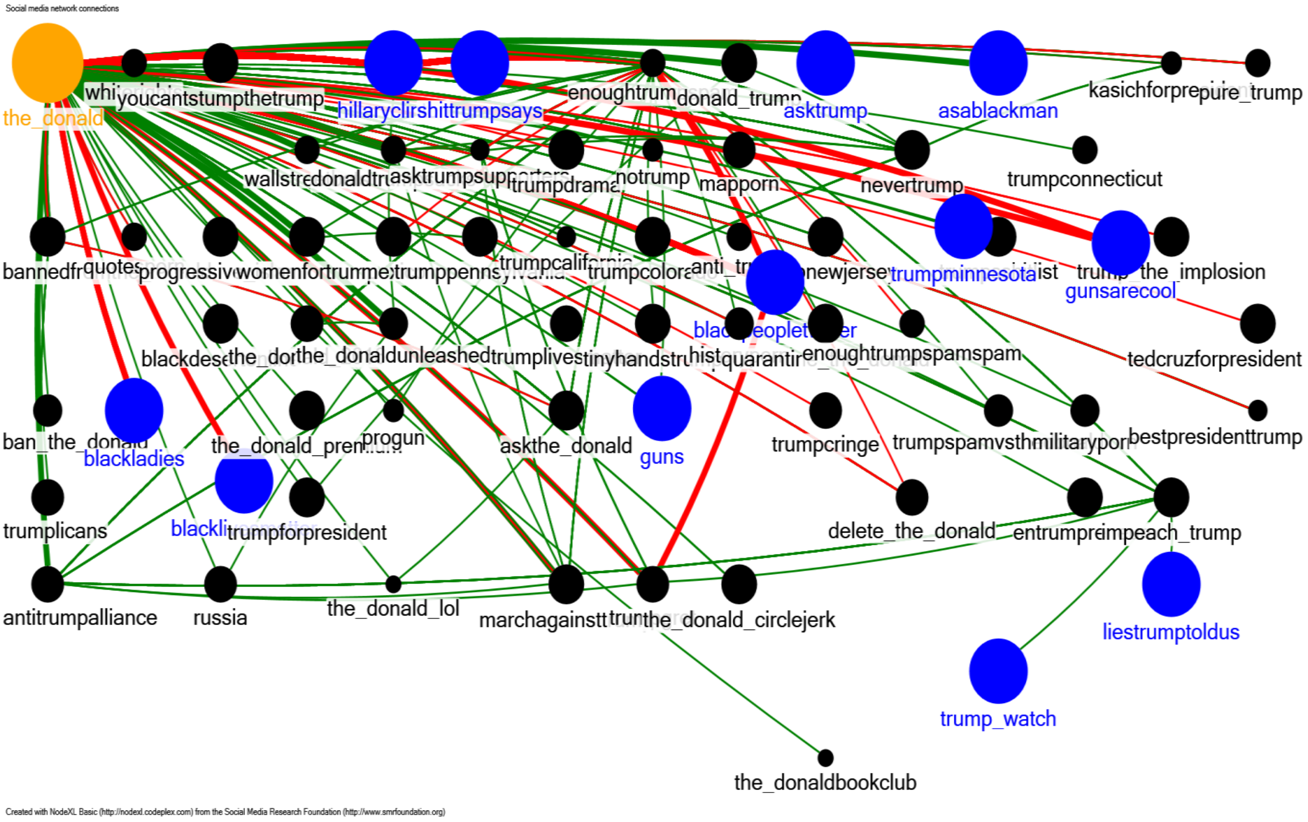


Fig. 1 Relationship between active subreddits and Donald Trump during 2016 US elections.

Dataset: soc-redditHyperlinks-body, soc-redditHyperlinks-title

Questions: Which were the most active subreddits? What were the subreddits with the most user engagement? and what were their relationships with Donald Trump related topics during 2016 US elections?

Insights:

1. From Fig. 1, we can observe the relationship between most active subreddits in US (shown by Blue and Yellow nodes) related to the Redditt topic 'Donald Trump' during the 2016 US elections.

2. The size of the nodes is proportional to the sum of the responses given by each subreddit. Therefore, the nodes with larger sizes like the blue nodes are the ones with most responses.

3. We can see that subreddits like *'shittrumpsays', 'asktrump', 'liestrumptoldus', 'hillaryclinton', 'gunsarecool', 'blackladies', 'askablackman', etc, are* the ones with most influence and user engagement.

4. Majority of the edges are converged towards the *'the_donald'* subreddit (represented by the orange node) making it the one with maximum number of inward edges/responses. Most of the responses by the influential subreddits shows a negative relationship with *'the_donald'* subreddit. This shows that at the time of 2016 US elections, most of the subreddits related Donald Trump to the propagated a negative sentiment in the community.

5. From the above we can see that the US 2016 elections was most influenced by users with strong interests in Racist groups like *‘askablackman’ ,* Users with strong opinions on Guns (as observed by the subreddit *‘gunsarecool’),* controvery promoting groups like *‘shittrumpsays”, ‘asktrump’, ‘liestrumptoldus’* and *‘the_donald’* and conspiracy promoting groups like ‘*usaexposed’.*

6. Hence from the above we can see that the 2016 US election was majorly controlled by conspiracy, racism, gun propaganda and viral fake news controversies.

Harmful Subreddits/Nodes identified:
‘shittrumpsays’, ‘asktrump’, ‘liestrumptoldus’, ‘hillaryclinton’, ‘gunsarecool’, ‘blackladies’, ‘askablackman’ etc.

Reasons for using *“Edge List”* data modelling technique for this visualization:
Here we had to plot the connection between the subreddits as well as the relationship linkages/edges linking those subreddits nodes. Hence because of the graphical nature of the data analysis problem, we processed the data using Edge List algorithm passed the data to graph visualization tool NodeXL. Here Edge List data modelling helps us maintain track of the numerous edge and vertices pair used for this visualization.

Plot 2: Relationship between active subreddits in US that are related to Crime.

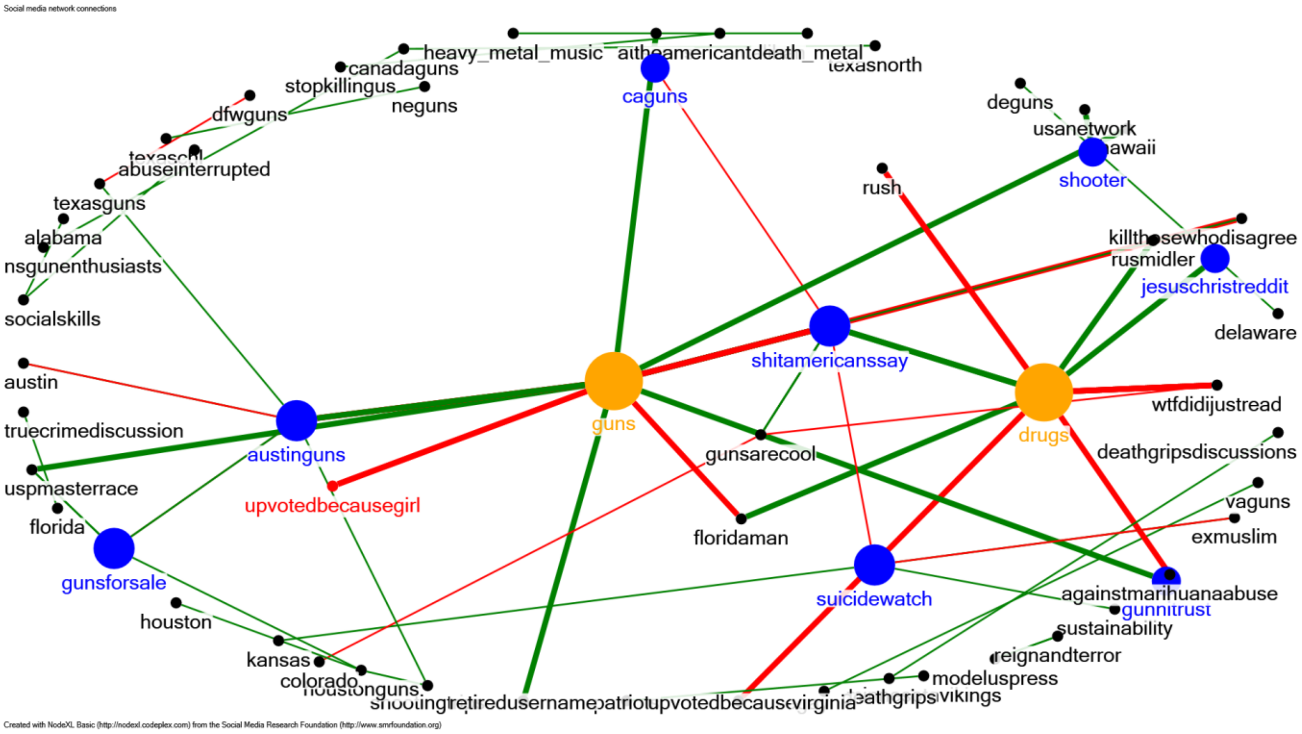


Fig. 2 Relationship between active subreddits in US that are related to Crime

Dataset: soc-redditHyperlinks-body, soc-redditHyperlinks-title

Question: What are the most active/ user engaged Crime related subreddit groups in USA? What are their underlying relationships? How did these groups behave during the 2016 Donald Trump US elections?

Insights:

1. From Fig. 2, We can see that the subreddit groups with the maximum user negarement/ user acitiy are the nodes in the graph with maximum number of edges ie the subreddit groups *'Guns' and 'Drugs'* (highlighted in yellow).

2. According to the graph, *'shitamericanssay', 'austinguns', 'jesuschristreddit', 'caguns', 'gunsforsale'* and *'suicidewatch'* are some of the other second most active subreddits and all of these are connected to either the *'Guns' and 'Drugs'* subreddit groups.

3. Both the groups *'Guns' and 'Drugs'* share an almost equal number of red and green edges representing people's positive and negative sentiments towards the groups. A visualization like this can be interpreted as result of the extremely polarizing environment Donald Trump had created during 2016 US elections. Dividing the people equally both for and against 'Drugs' and 'Guns' related issues in USA.

Harmful Subreddits/Nodes identified:
'guns', 'gunsforsale, 'drugs', 'suicidewatch', 'shooter', 'caguns', 'austinguns', 'shitamericanssay', 'killthosewhodisagree' etc

Reason for using *"Edge List"* data modelling technique for this visualization:
Here again, we had to plot the connection between the subreddits as well as the relationship linkages/edges linking those subreddits nodes. Hence because of the graphical nature of the data analysis problem, we processed the data using Edge List algorithm passed the data to graph visualization tool NodeXL. Here Edge List data modelling helps us maintain track of the numerous edge and vertices pair used for this visualization.

Plot 3: Most influential subreddits w.r.t US Politics

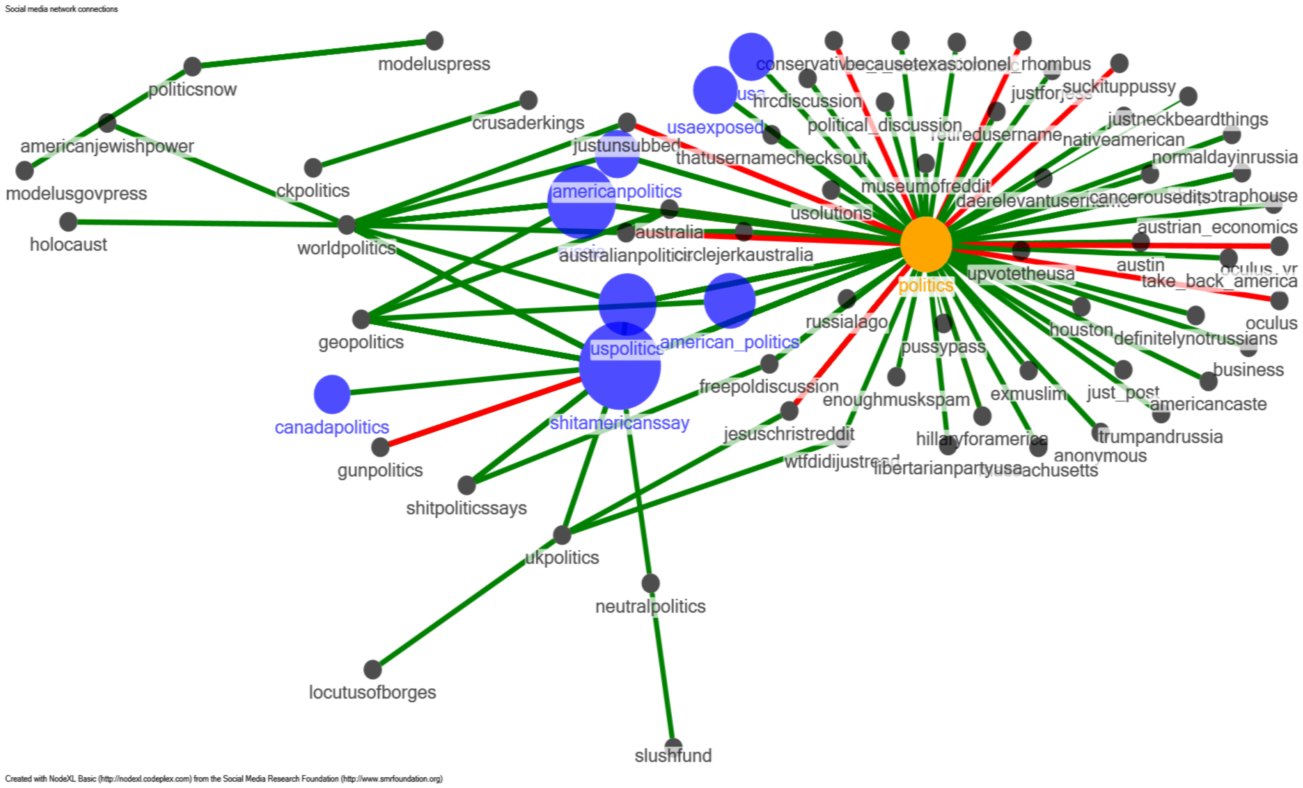


Fig. 3 Graphical representation of active subreddits towards US Politics

Dataset: soc-redditHyperlinks-body, soc-redditHyperlinks-title

Questions: Which subreddits are the most influential to the US politics event? What is the relationship between them and other active subreddits?

Insights:

1. From the graphical representation in Fig. 3, we can observe the relationship between several active subreddits in the US and their relationships with the US Politics event.

2. Majority of the subreddits(nodes) are directly or indirectly related to the subreddit 'politics' (shown by the orange node) which is visible by the edges converging at the 'politics' node. Among these majority, some of the most influential subreddits are represented by the blue nodes.

3. We can see that the subreddits like '*shitamericanssay', 'usaexposed'*, *'uspolitics'* and *'canadapolitics'* are the most influential subreddits towards US Politics event.

4. Through this we can see that the 2016 US elections was primarily dominated by conspiracy and toxic inflammatory content and fake news promoting subreddit groups like '*shitamericanssay'* and *'usaexposed'.*

5. We can also see that Redditt users from neighboring countries like Canada were also heavily engaged in influencing 2016 US elections, as observed from the subreddit group *'canadapolitics'.*

Harmful Subreddits/Nodes identified:

'gunpolitics', 'usaexposed', 'take_back_america, 'russialago', 'shitamericanssay', 'justunsubbed', etc

Reason for using *"Edge List"* data modelling technique for this visualization:

Here again, we had to plot the connection between the subreddits as well as the relationship linkages/edges linking those subreddits nodes. Hence because of the graphical nature of the data analysis problem, we processed the data using Edge List algorithm passed the data to graph visualization tool NodeXL. Here Edge List data modelling helps us maintain track of the numerous edge and vertices pair used for this visualization.

Plot 4: Sentimental Analysis of the top 10 active subreddits related to US elections / Donald Trump.

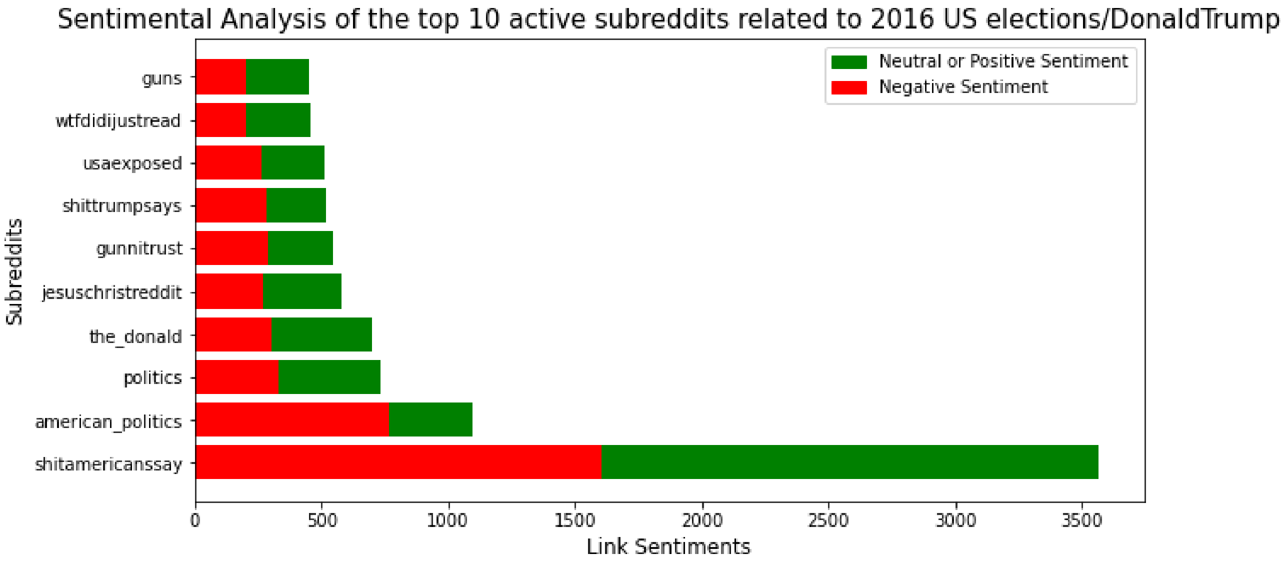


Fig. 4 Sentimental Analysis of the top 10 active subreddits related to 2016 US elections / Donald Trump

<u>Dataset</u>: soc-redditHyperlinks-body

<u>Questions</u>: What were the top 10 active subreddits related to US elections / Donald Trump? What was the distribution of peoples' sentiments within these group themselves?

<u>Insights</u>:

1. Fig. 4 represents a stacked horizontal bar graph of the most active top 10 subreddits related to US elections / Donald Trump and their sentimental analysis.

2. We can observe that the subreddit *'shitamericanssay'* was the most active with maximum number of positive and negative sentimental responses.

3. From the above we can see that the US 2016 elections was most influenced by users with strong interests in Christian religion (as observed by the subreddit '*jesuschristreddit'),* Users with strong opinions on Guns (as observed by the subreddit '*guns'),* controvery promoting groups like *'shitamericanssay'* and *'the_donald'* and conspiracy promoting groups like '*usaexposed'.*

4. Hence from the above we can see that the 2016 US election was majorly controlled by conspiracy, religion, gun propaganda and viral fake news controversies.

<u>Reason for using *"Adjacency List"* data modelling technique for this visualization</u>:
Due to high number of edges and vertices, we used Adjacency list modelling technique to fetch and process data for this visualization because it is significantly more space-efficient than the other modelling techniques.