Report

Assignment 3: Graph Data

Reddit Data Analyst

CSC 501

University
of Victoria

Submitted to:

Prof. Sean Chester

Submitted by:

Anushka Halder (V00961967)

Sanjana Arora (V00966221)

Tavanpreet S. Oberoi (V00963163)

1. Introduction

Reddit is a network of communities where people can dive into their interests, recreation, hobbies, and passions. It has dedicated groups known as Subreddits where people write about topic they commonly are interested in. Subreddits are denoted by /r/election, where people write anything related to elections.

Because it is a website with over 330 million members, it is worthwhile to examine the posts in subreddit that are directed at other subreddits to have a better understanding of the link sentiment between the subreddits.

As said, reddit is a network of communities, hence data needs to be network modelled to identify the relationships between different subreddit groups based on different parameters and sentiments.

DATASET: We have used two tab-separated-values (TSV) files including the Subreddit Hyperlink Network extracted from the title of the post and the body of the post. These files include information relating to source and target subreddit community of a hyperlink along with temporal details such as 'TIMESTAMP', post details like 'POST_ID' ,' LINK_SENTIMENT', word count, etc. We also used the Subreddit embedding dataset capturing the embedding vectors representing each Subreddit.

All the team members brainstormed on the theme of this Reddit data analysis and developed the questions based on the theme that this report would answer. As the dataset is huge, it was important to create a data model that has the most optimized memory and space requirements. Further, based on the kind of analysis we wanted to achieve, a conceptual and logical model that would best support the analysis is designed and discussed in further sections of the report.

THEME OF ANALYSIS: We will be conducting analysis of the general relationship between Subreddits based on different set of post properties such as Religion, indicators of negative sentiments such as anger, dissent, swear, etc. We identify the Subreddits having the most positive and negative relationships. Further, we also try to validate the' LINK_SENTIMENT' attribute that has been included in the data. We validate the same by creating our own indicator that indicates whether a post is having negative or positive sentiment based on the post properties. Additionally, we explore the increase or decrease in popularity of a Subreddit based on certain events such as elections or sports.

This report captures discussion on conceptual and logical data models, keeping the scalability factor in the process.

2. Data Modelling

2.1 Conceptual Data Model

The given dataset of subreddits contains relationships having properties between two or more subreddits in the form of spreadsheet (relations). It is evident that Relational Database Management System (RDBMS) can be implemented, but with such a directed social network dataset where the goal is to lookup relationships between different entities, RDBMS can be inconvenient in terms of space and time requirements.

Why Graphical Database?

Graphical database is much more convenient in case of operations such as looking up a node A and inserting a new subreddit community B who is targeting node A; as it would just require searching for a node with id A and add links with B rather than traversing through rows or array of nodes. Hence, storing the data in form graphical database provides advantages in common operations such as insertion, deletion, creation, analysis amongst different nodes of a graph.

Proposed Graphical Model

The below diagram represents the conceptual model used for the Reddit dataset. Each record in dataset captures information relating to the name of originating community of a hyperlink of a post ('SOURCE_SUBREDDIT') and the target community ('TARGET_SUBREDDIT'). Since, the questions that we are asking of the data mostly relate to finding out the type of **relationships** between two or more communities **(entities)** who are posting on the subreddits, we decided to model the names of attributes **'SOURCE_SUBREDDIT'** and **'TARGET_SUBREDDIT'** as the vertices 'V' of our graph G. A hyperlink *originated_from* 'SOURCE_SUBREDDIT' and *has_targeted* community 'TARGET_SUBREDDIT', therefore the vertices **'SOURCE_SUBREDDIT'** and **'TARGET_SUBREDDIT'** relate to each other by a Reddit post with a post id and hence, making the edges E of the graph G. Further, this indicates a direction associated with each post, so we have created a directed labeled graph wherein, the properties of interest such as 'liwc_anger', 'liwc_we', 'liwc_you', 'liwc_they', 'liwc_negate', 'liwc_affect', 'liwc_posemo', 'liwc_negemo', 'liwc_anx', 'liwc_anger', 'liwc_sad', 'liwc_relig', 'liwc_death',' liwc_assent' and 'liwc_dissent' are used as labels for each of the edges.

These edges are weighted +1 and -1 based on the 'LINK_SENTIMENT' attribute wherein, +1 indicates a positive sentiment of the post and -1 relates to a negative sentiment of the post. Since, each 'SOURCE_SUBREDDIT' and 'TARGET_SUBREDDIT' community shares multiple Reddit posts with each other, this would clutter a lot when visualized, for simplification, we have taken an average of each of the property values for positively weighted edges and negatively weighted edges respectively to represent a single edge between multiple vertices.

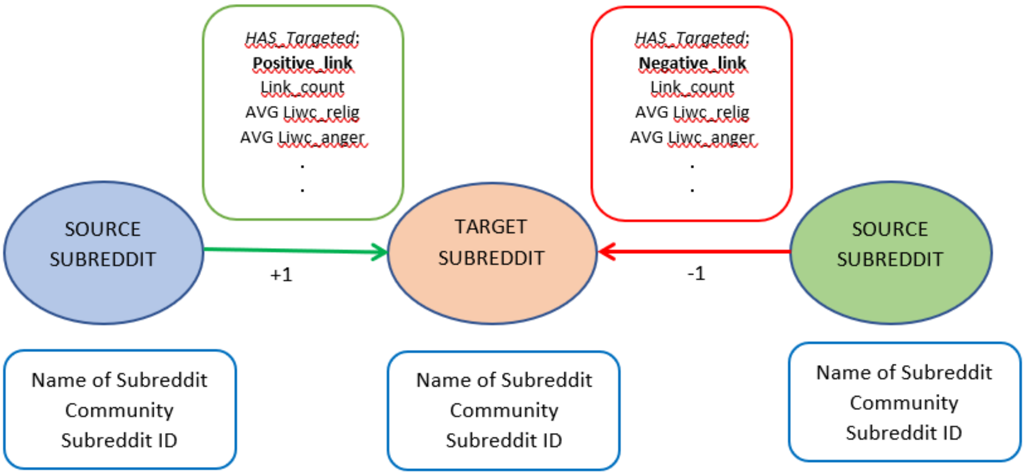


Fig.2.1.1 Conceptual model for Reddit Dataset

Hence, the conceptual model can answer questions like: -

1. **Who received the most negative sentimental posts?**
   Ans. Since the edges are weighted; we can identify the node receiving the highest number of negative sentimental posts by counting the number of edges pointed towards a node as captured in Plot 1.

2. **Which communities discuss religion the most?**
   Ans. As edges capture the 'liwc_religion' label, we can identify the edge having the maximum 'liwc_religion' label value over all the edges as captured in Plot 4.

3. **Which communities exchange the most positive posts?**
   Ans. We can count the positive weighted edges between two nodes to find the two nodes exchanging the most positive posts as captured in Plot 3.

2.2 Logical Data Model

Logical Data Model defines how system must be implemented while adding details about data structure being used. As it has been discussed in Conceptual Data Modeling section that given dataset is modeled as a directed Graph. So, all the below analysis is done keeping directed Graphs in understanding.

The three common data structures used with graph data, each having its own advantages over other are:

- Edge List
- Adjacency Matrix
- Adjacency List

Edge List: An edge list is a list or array that contains all the edges in a graph. Edge lists are one of the simplest ways to represent a graph. The basic data structure for keeping track of all nodes (also known as vertex) represented using |V| and edges represented using |E| or (v, w) in this approach is a single list of pairs represented below.

edge_list -> [e1, e2, e3, e4, …]

which can also be represented as [(v1, v2), (v2, v3), (v3, v1), (v3, v4) …]

where, v ∈ V for all nodes in graph G and e ∈ E for all edges in graph G.

Below is the sample representation of the given data:

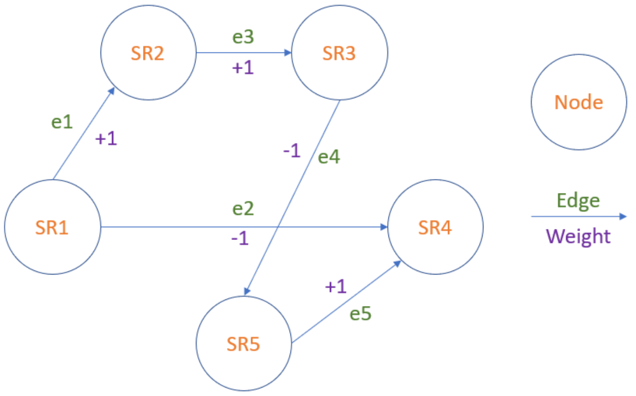


Fig 2.2.1. Graphical Representation of given Data

Database Representation using Edge List

In our dataset, 'Source_SubReddit' and 'Target_SubReddit' are represented as Vertices (|V|) and each row of data is represented as Edge (|E|). The 'Link_Sentiments' which can be positive (+1) and Negative (-1) are represented as weights of the edge E in the graph G. The above representation of the data can be represented in edge list as:

Edge_list = [(SR1, SR2, +1), (SR1, SR4, -1), (SR2, SR3, +1), (SR3, SR5, -1), (SR5, SR4, +1)]

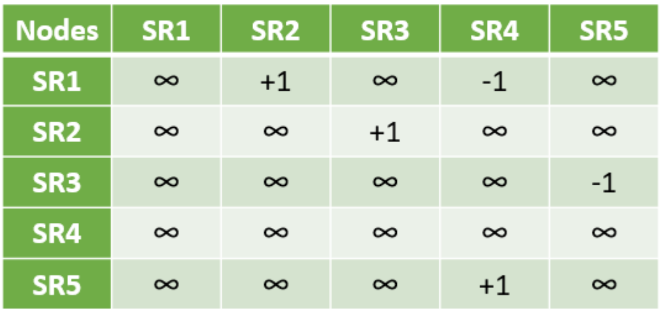
Adjacency Matrix: is a square matrix used to represent a finite graph. As the name represents, it is primarily used to identify adjacent nodes in the graph. Adjacency Matrix is a 2D array of size V x V where V is the number of vertices in a graph. Each vertex is represented against all other vertex forming the square matrix. If there exists the relationship between two vertices (v, w) forming an edge (e1) in Graph G, then it can have following properties in case of non-weighted graphs:

- Undirected Graph: It represents the symmetric matrix because for an edge between v and w, the matrix at **both** the position([v][w], [w][v]) will be populated with value 1. If there is no edge between nodes, then those positions are populated with 0.
- Directed Graph (Our case): If there is an edge from vertex u to vertex w, then **row u and column w of the matrix** is populated with 1 and rest with 0.

In case of weighted graphs, value where an edge exist can be replaced with weight while other with infinity (∞).

Database Representation using Adjacency Matrix

The representation of Fig 1. in adjacency matrix is represented in Fig 2 below.



Nodes	SR1	SR2	SR3	SR4	SR5
SR1	∞	+1	∞	-1	∞
SR2	∞	∞	+1	∞	∞
SR3	∞	∞	∞	∞	-1
SR4	∞	∞	∞	∞	∞
SR5	∞	∞	∞	+1	∞

Fig 2.2.2 Adjacency Matrix

Adjacency List: is an array of linked lists where the size of the array is equal to the number of vertices in the graph G. The index of the array represents a vertex and elements in linked lists represents other vertices that form an edge with the vertex. It is combination of Adjacency Matrix and Edge List. Let the array be an adj_list[]. An entry adj_list[i] represents the list of vertices adjacent to the **i**th vertex (edge). The weight of the edges can also be represented as list of pairs.

Database Representation Adjacency List

The adjacency list of Fig 1. can be represented as
SR1 – [0] → [[SR2, +1], [SR4, -1]]
SR2 – [1] → [[SR3, +1]]
SR3 – [2] → [[SR5, -1]]
SR4 – [3] → []
SR5 – [4] → [[SR4, +1], [SR4, -1]]

Comparison of different available Graphical Databases?

Properties	Edge List	Adjacency Matrix	Adjacency List
Space	O(|E|)	O(|V|^2|)	O(|V|+|E|)
Vertex Insertion	-	O(|V|^2|)	Best
Edge Insertion	Best	Better	Good
Querying to check an Edge	O(|E|)	O(1)	O(|V|)
Removing an Edge	O(|E|)	O(1)	O(|V|+|E|)
Outdegree	Good	Better	Best

Based on above table following points are considered while modelling the given dataset:

- **Adjacency Matrix:** The dataset (body dataset) has 51278 unique nodes (vertices) which requires a lot of space and since, adjacency Matrix is not an efficient memory solution for the data with so many vertices, this method of storing the data is not preferred. Further, it does not support the scalability of the model, as Vertex Insertion is O(|V|^2).
- **Edge List:** The dataset (body dataset) contains total of 286561 edges, thereby increasing space complexity. Also querying to check an adjacent node (primary goal of the modelling) is O(|E|) which further increases with more edge insertion.
- **Adjacency List:** It is the best suited model among all three, though it has some drawbacks, but that does not affect much as primary purpose is to query to check an adjacent node

while saving the memory. It also supports scalability, as vertex insertion is best comparing to adjacency matrix, compromising on edge insertion.

Performance of all the three models is also compared and it was found that adjacency list outperforms others on major parameters.

Chosen Logical Model based on Neo4j (Whiteboard friendly Graphical Modelling Database)

As discussed in the conceptual model, we decided to model the data in form of a directed labeled graph; we also explored the Neo4j database. Neo4j was a preferred graphical database as it is known as "whiteboard-friendly", and we could easily model our data in the same form that was discussed in conceptual model. Since, Neo4j allows storing, managing, and visualizing the graph data in its more natural and connected state, we were able to visualize relationships between nodes by involving edges with a greater number of relationship properties such as number of reddit posts, average value of 'liwc_anger', 'liwc_swear', 'liwc_anx', etc. Further, as captured in the further sections of report, we were able to run complex queries for forming subgraphs in a lighting fast manner.

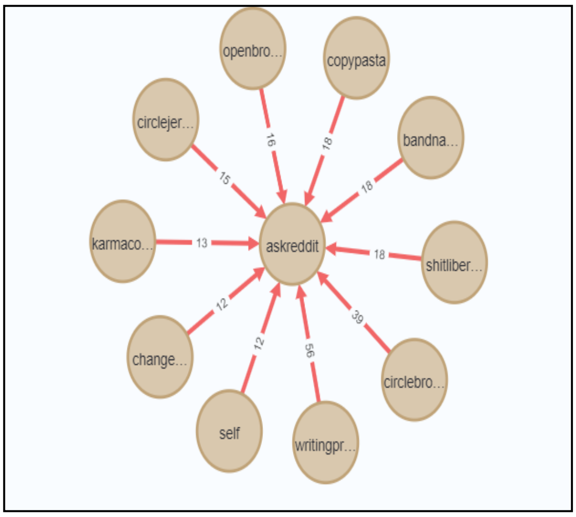
The following process was followed for modeling the data in Neo4j:

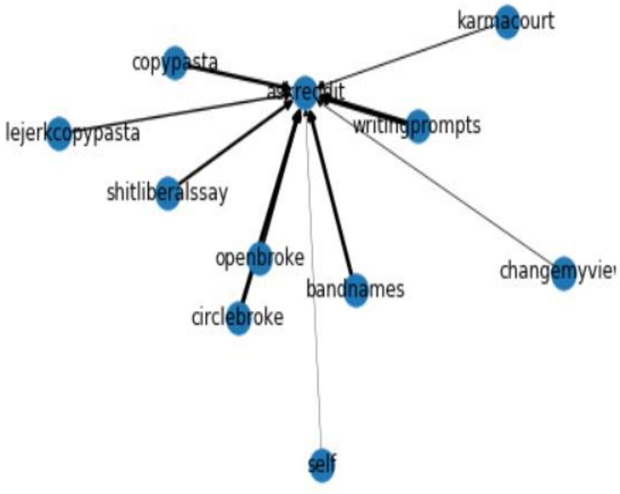
- The .tsv files were converted to .csv files that could be used in Neo4j. A new database was created using the .csv files.
- A unique community subreddit name constraint was added to the database.
- A graph having the subreddit communities (entities) as nodes was created. *The graph G is of order 51278 as there are 51278 nodes in the graph.*
- The nodes of the graph were connected to each other by adding the edge weights relating to the positive and negative link sentiments.
- All the edges were aggregated with each other by taking the count of number of positive sentiment links and negative sentiment links, along with an average of properties such as 'liwc_anger', 'liwc_anx', ''liwc_swear' etc.
- Multiple subgraphs were created by querying the created graph database to derive the required insights.

3. Exploratory Data Analysis

PLOT-1: Target Subreddit receiving most negative posts

We first found the subreddit receiving the maximum negative sentimental post by *counting* the number of edges with -1 weight constructed between two subreddits. Following this, we calculated the top 10 subreddits that targeted askreddit (identified as the *highest targeted subreddit*).



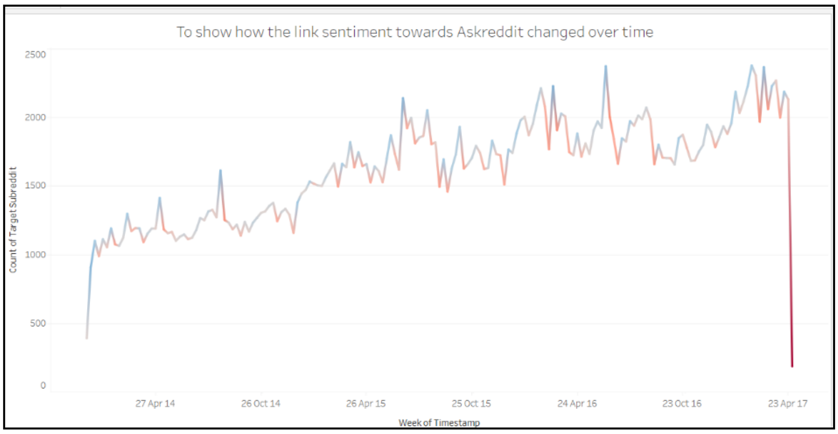


INSIGHTS- In above visualization, posts published in source subreddit with negative sentiments has referred to the 'Askreddit' the maximum times. The values mentioned in the edges is the count of negative link sentiments. Askreddit is one of the most popular subreddit having the most consistently exciting, absorbing, and cringe-worthy posts thereby getting targeted with the greatest negative comments.

PLOT-2: Sentiment change towards Askreddit

We plotted the line chart between years vs the average sentiment of subreddits towards the leading subreddit by using Tableau.

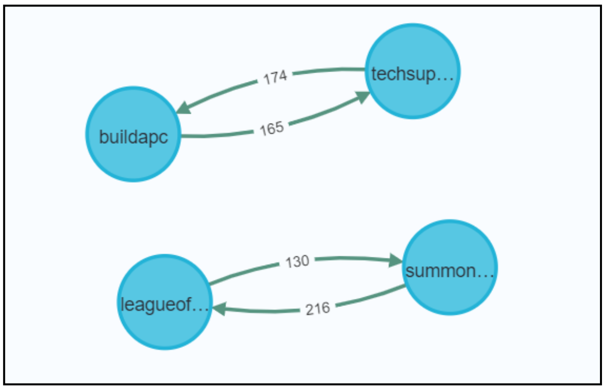
INSIGHTS – The below line chart shows the trend of change in the sentiment towards askreddit over the years with shades of blue showing that there were more positive links shared during that time and red showing negative sentiments. We can see that in the beginning positive links have been shared however the trend changed over the years ending in more negative posts shared by different subreddits. This can confirm our above analysis showing that askreddit has the most negative sentiments.



PLOT-3: Subreddit pairs with the maximum positivity exchange

Two pairs are found where the source and the target subreddit has posted *maximum* positive links between each other. We achieved this by finding the *sum* of links between the two subreddits with the *maximum* positive posts.

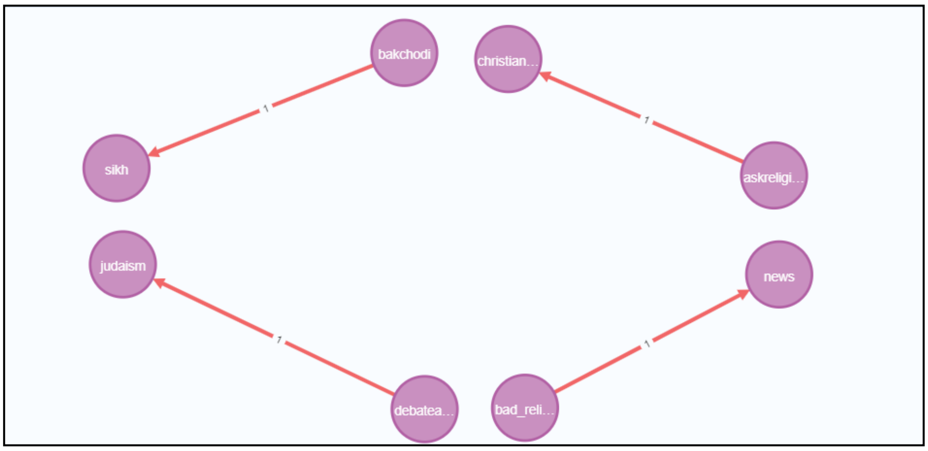
INSIGHTS – It has been noticed that 'Buildapc' and 'techsupport' subreddits groups provide help to people who are in the field of *hardware computing* and need guidance to build computer parts. While 'Leagueoflegends' and 'summonersschool' provide resources and help to the LeagueofLegends (football) team players to improve their game. It can be safely concluded that where the knowledge has been shared about a specific domain has the most positive link sentiment.



PLOT-4: Subreddit pairs with the most negative religious link shares

The top four pairs with the *most* negative discussions on religion by *counting* the number of links between the subreddits (on property religion) have been visualized in *descending* order.

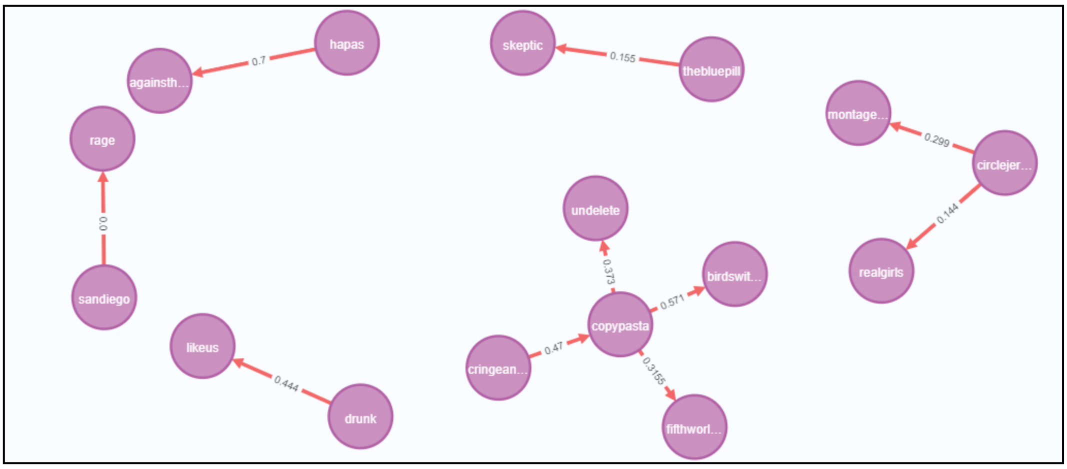
INSIGHTS – It has been observed that the most subreddits groups talks about or belongs to the Sikh, Christianity, and Judaism religion. We can verify the observation by looking at the names mentioned on the nodes which indicates that all of them have discussed negatively about a specific religion or posted links on religious topics.



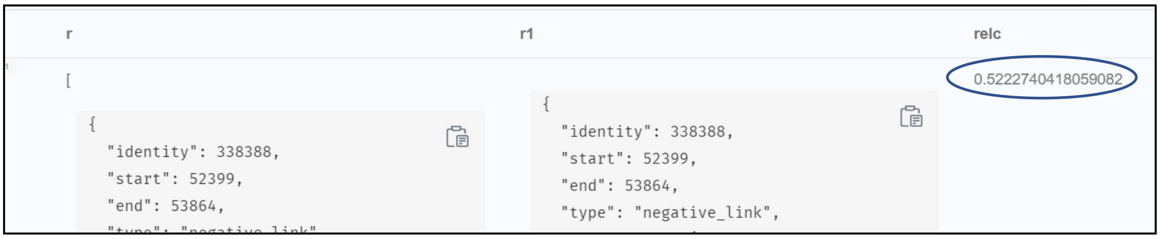
PLOT-5: Top subreddit pairs with the most negative link shares

We have the top pairs with the most negative discussions based on the cumulative *sum* of the links shared on various negative topics like anger, sad, anxiety, death, dissent, negative emotions. We found the *sum of the averages* of these properties and presented the node links with the highest value.

INSIGHTS – We found the top groups where the maximum negative links are shared based on numerous negative topics. The edges have the average negative sentiment value mentioned. There have been discussions about topics like ill-treating/slaughtering animals, how covid is a conspiracy, racist comments on different skin color and so on. It can be said that these pages can be easily identified based on their property values.



To validate the link sentiment attribute of the dataset, we also found the *cumulative sum* of the negative properties where positive sentiments were shared among the subreddits. It was observed that the sum of properties relating to negative emotions in the negative sentimental edges is much higher than the positive sentiment (circled in figures below) highlighting the fact **that we can rely on the link sentiment attribute to locate a subreddit for a negative/positive post.**

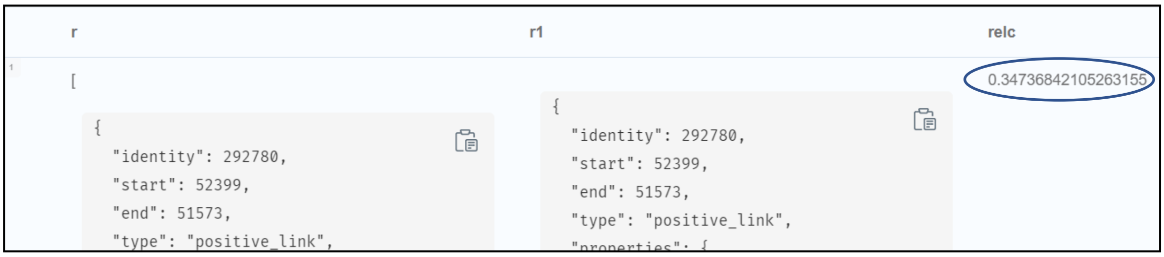


r	r1	relc
[		0.5222740418059082

```
[
  {
    "identity": 338388,
    "start": 52399,
    "end": 53864,
    "type": "negative_link",
```

```
{
  "identity": 338388,
  "start": 52399,
  "end": 53864,
  "type": "negative_link",
```

Negative properties with negative sentiment



r	r1	relc
[		0.34736842105263155

```
[
  {
    "identity": 292780,
    "start": 52399,
    "end": 51573,
    "type": "positive_link",
```

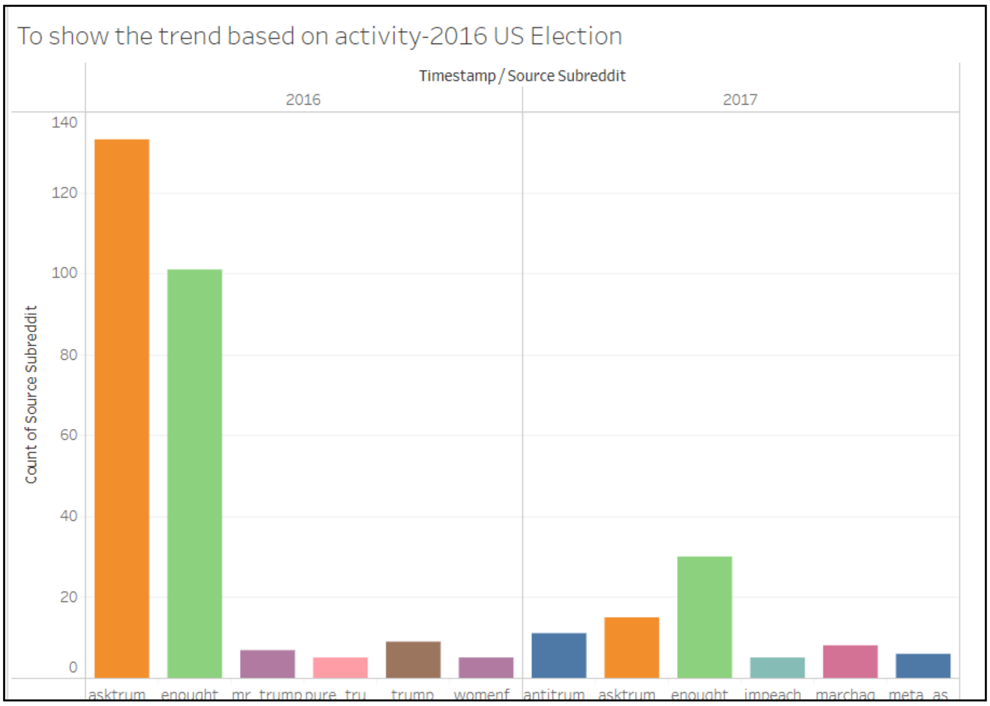
```
{
  "identity": 292780,
  "start": 52399,
  "end": 51573,
  "type": "positive_link",
  "properties": {
```

Positive properties with negative sentiment

PLOT-6: <u>Trend of a specific subreddit based on events</u>

We plot a bar chart between different subreddits relating to Donald Trump and found the count of negative links posted towards them in 2016 and 2017.

INSIGHTS – The visualization shows that there has been higher number of links shared relating to Donald trump during 2016 Elections and have drastically decreased in the next year. We can conclude that some pages are active only during a major event such as elections and become inactive over the period as the events get old.



Conclusions from visualizations:

As discussed in conceptual and logical models, we created a graphical database using neo4j for the reddit dataset. The edge labels facilitated us to create subgraphs using cypher query language for our visualizations. We used different functions and operations like collect, sum, size etc. to find the edges between the nodes having the required values. We first found out the subgraph with the node that has the maximum incoming edges with the negative sentiment and then observed the sentiment change of that node (Askreddit) over the years. The data model also helped us validate the 'link sentiment' attribute analysis by cumulating the values of all the negative properties (anger, death, sad, dissent and negative emotion) in the dataset. Another validation was done on the religion-based visualization (plot 4) where we confirmed that the names of the subreddits confirm that all the discussions were based on religion. We finally closed our analysis by understanding how any event can affect the activity or popularity of a subreddit showing the example of US Elections. At the end of the analysis, we concluded that our model could rightly answer the questions.

4. Scalability

- **Processor** – Intel(R) Core (TM) i7-4510U CPU @ 2.00GHz 2.60 GHz
- **RAM** – 8GB
- **Tool/software** – Jupyter Notebook, Neo4J, Tableau and Google Colab
- **Language** – Python 3.7 and Cypher

In the above report, working on the primary goal of identifying the relationships between different subreddits and how those are interlinked in the network based on different parameters are studied.

Properties time complexity	Edge List	Adjacency Matrix	Adjacency List	Neo4j
Node and Edge Creation	524 ms (Unlabeled relation)	63000 ms (Unlabeled relation)	40600 ms (Unlabeled relation)	139946 ms (Labelled relation)
Querying to find and visualize top 3 negative subreddits	174 ms	495000 ms	2790 ms	10 ms
Plotting Maximum numbers of negative feedbacks based on Adjacency List	77 ms	-	62 ms	55 ms

It has been observed that based on the goal, dataset has been graphically modelled to take advantage of the networks being formed between multiple subreddits. It has been theoretically discussed in the logical modeling section about why we think Adjacency list would perform better over other graphical data structures (Edge List and Adjacency Matrix) in the given problem domain. This is evidently being proved with the execution and querying time provided in the table above as well. It also supports in the scalability of the model as inserting new vertex (new subreddit groups) and new edges (newer data) in Adjacency List is much faster as compared to other models. As there is no need of deleting an edge, once the data has been modelled, we do not need to bother on poor performance of the adjacency list on that parameter.

Why Neo4J?

Neo4j provided a seamless and scalable solution to creating and managing a graph database. It also provided the flexibility to visualize the multiple subgraphs for representing the insights. As it can be seen from Plot 1 in EDA section, that Neo4j visualizations are more interesting and self-explained while it is difficult to do with existing graphical data structures. Also, there are limited libraries available in different languages which supports graphical representation, hence Neo4j is overall preferred with graphical dataset. It has also been proved in the table above that performance of Neo4j is better than standard graphical databases. Querying and modelling the labelled database with weights can be easily done using Neo4J Cypher (Querying language). The insertion time into Neo4j is considerably more because we included labelled nodes and edges which is not done using standard graphical data structures (Adjacency List, Adjacency Matrix and Edge List).

Further, Neo4j has the capability to connect with Apache Spark APIs and hence it can support the streaming data, incase this model is used for real-time (continuous) data in future.