Submission report : CSC 501 :201909

Assignment 3 : Team D

Team members :

Sushil Paneru V00936990
Rajneesh Gulati V00949939

This report contains the insights, design choices, data visualizations, challenges and adherence details to the rubric provided for this assignment.

Table of contents

Data Models and Design Choices

1. Extract Phase

For the extraction of data from the tsv file, we directly used Neo4j. Neo4j provides rich set of features to extract data from csv, tsv files, etc.

2. Transform Phase

As part of transformation phase, we converted the data given to us into nodes and relationship to form a graph model of the data.

a. **Creation of subreddit, post nodes**
   For each row, the source and target subreddit column was used to create subreddit nodes and post id was used to create post nodes. The subreddit node contains only the id of the subreddit where as the post node contains link sentiment, text properties. The relationship between these nodes can be shown as **sr:Subreddit-(target_by)->p:Post-(target_to)->tr:Subreddit.**

b. **Creation of hour, year and day nodes for extracting temporal insights**
   For every cross-link amongst subreddits, we have timestamp of the event. One possible way to model the graph would be to include the timestamp in edges that connect the subreddits. Doing so, hampers the performance of the cypher queries which are seeking insights related to time because of the fact that any time related query would have to iterate over all the edges. But if we create nodes for day, hour, year, we only have to traverse over selected edges.

   For example,
   Traversal via temporal nodes:

   ```
   MATCH (hr:HourNode)-[:POSTED_AT]->(p:POST) where hr.id = 10 RETURN p.id
   ```

   Run time: 33ms

   Traversal of all edges:

   ```
   MATCH (sr:SubReddit)-[t:targets]->() where t.timestamp.hour = 10 RETURN sr.id
   ```

   Run time: 323ms

   Performance improves by a factor of 10.

c. **Creation of redundant edges between cross-linked subreddits (Denormalization)**

Even though post node contains all the information related to a post that cross-links, we also added a direct edge (named :targets) from one subreddit to another for every cross-link event (graph model is shown below - fig 1). The reason for doing so is to reduce the number of hops especially for non-temporal scenarios.

For example, the case where query extracts subreddit pairs where cross-linked post has more than 100 characters

1. Traversing via redundant edge that links subreddits directly
   MATCH (sr:SubReddit)-[t:targets]->(tr:SubReddit) where t.textProp[0] > 100 RETURN sr.id, tr.id
   Runtime - 2768ms

2. Traversing via post node
   MATCH
   (sr:SubReddit)-[:target_by]->(p:POST)-[:target_to]->(tr:SubReddit)
   where p.textProp[0] > 100 RETURN sr.id, tr.id
   Runtime - 4649ms

The first query is almost twice as fast as second query because it can be easily seen that second query traverses twice the number of edges to extract the same information. Even though the edge(:targets) looks redundant, it helps to reduce the number of edge traversal. **Thus, for temporal insights, path involving temporal node, post node is used whereas for non-temporal insights denormalized edge (:targets) is used.** Also, we are more concerned about OLAP operation, so having redundant edge should not have side effects.

d. **Creation of timestamp object**

The timestamp string for every post is converted to Neo4j's timestamp object which pre stores data related to day, hour, year and allows constant time data access.

e. **Indexes on ids**

Index on ids for the subreddit, post node and (:targets) relationship was created so that queries involving search on ids can be made performant. Doing so, helped to speed up the creation graph by a significant factor.

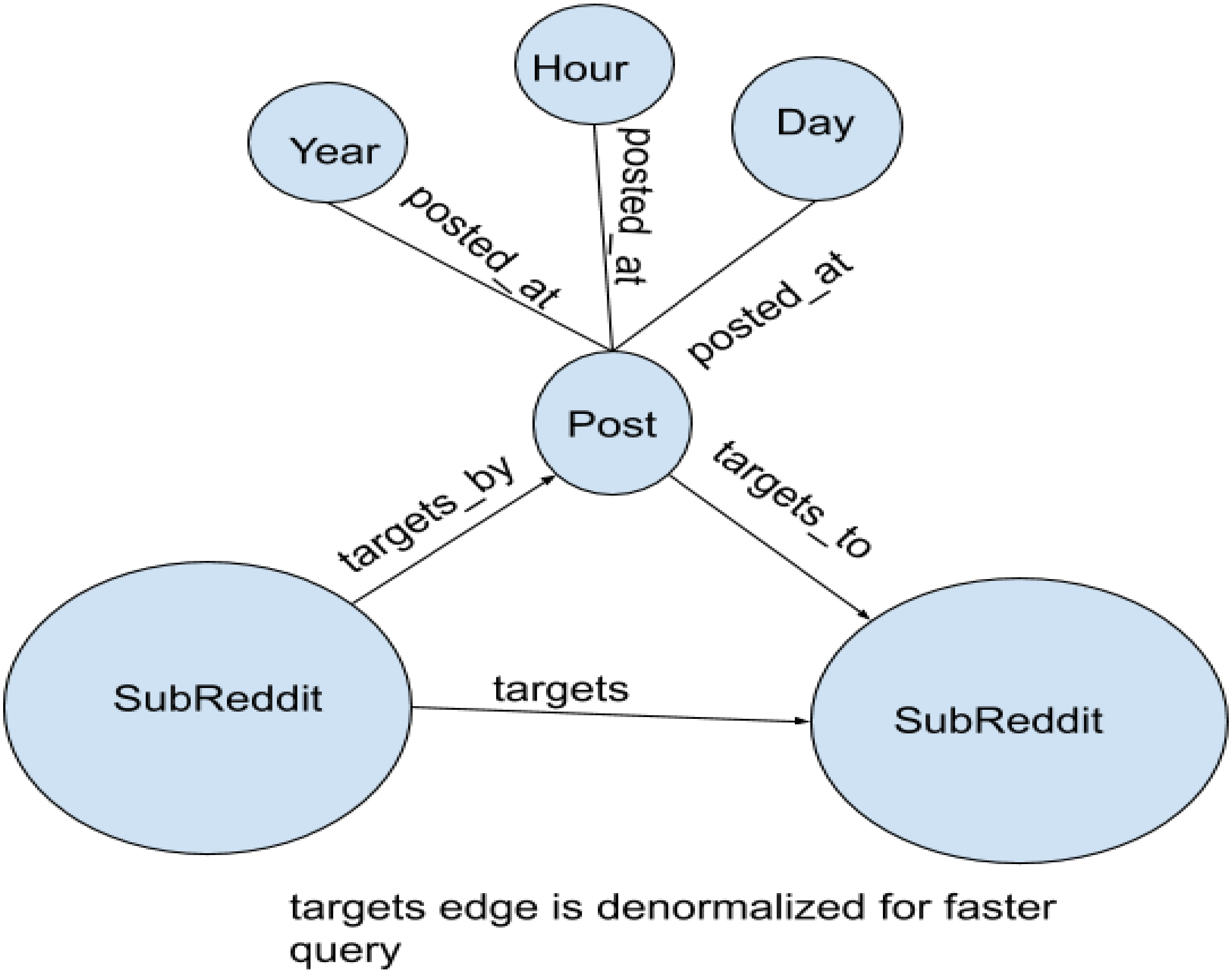


Fig. 1

3. Load phase

Transforming the whole data to knowledge graph and loading it to Neo4j caused a huge problem because the memory (RAM) was not sufficient. Doing so caused OOM exception time and time again. Thus, to solve this issue, we analysed the query plan and found that while creating relationship Neo4j uses a technique called **eager loading** which basically involves loading all the nodes in memory while creating relationship between them. This happens when we create nodes and relationship in the same query so, the trick is to avoid eager loading by separating load query for nodes and relationship. Hence, we were able to load the whole graph (involving both body and title) into the Neo4j's database within 3 minutes.

Load queries:

1. **Create all nodes in one query**

```
USING PERIODIC COMMIT 1000 LOAD CSV WITH HEADERS FROM
'file:///soc-redditHyperlinks-merged.tsv' AS row FIELDTERMINATOR '\t'
MERGE(sr:SubReddit{id: row['SOURCE_SUBREDDIT']}
MERGE(tr:SubReddit{id:row['TARGET_SUBREDDIT']})
MERGE(p: POST{id: row['POST_ID'], sentiment:
toInteger(row['LINK_SENTIMENT']),textProp: [x in split(row['PROPERTIES'],',') |
toFloat(x)]})
MERGE(hr: HourNode{id: datetime({ epochMillis:apoc.date.parse(row['TIMESTAMP'],
'ms', 'yyyy-mm-dd HH:mm:ss')}).hour})
MERGE(yr: YearNode{id: datetime({ epochMillis:apoc.date.parse(row['TIMESTAMP'],
'ms', 'yyyy-mm-dd HH:mm:ss')}).year})
MERGE(yr: DayNode{id: datetime({ epochMillis:apoc.date.parse(row['TIMESTAMP'],
'ms', 'yyyy-mm-dd HH:mm:ss')}).dayOfWeek})
RETURN count(sr)
```

2. **Create relationships separately**

```
USING PERIODIC COMMIT 1000 LOAD CSV WITH HEADERS FROM
'file:///soc-redditHyperlinks-merged.tsv' AS row FIELDTERMINATOR '\t'
MATCH(sr:SubReddit{id: row['SOURCE_SUBREDDIT']})
MATCH(tr:SubReddit{id:row['TARGET_SUBREDDIT']})
MATCH(p: POST{id: row['POST_ID']})
MERGE(sr)-[:target_by]->(p)-[:target_to]->(tr)
RETURN count(sr)
```

Similarly, other relationships like targets, posted_at is created in separate queries.

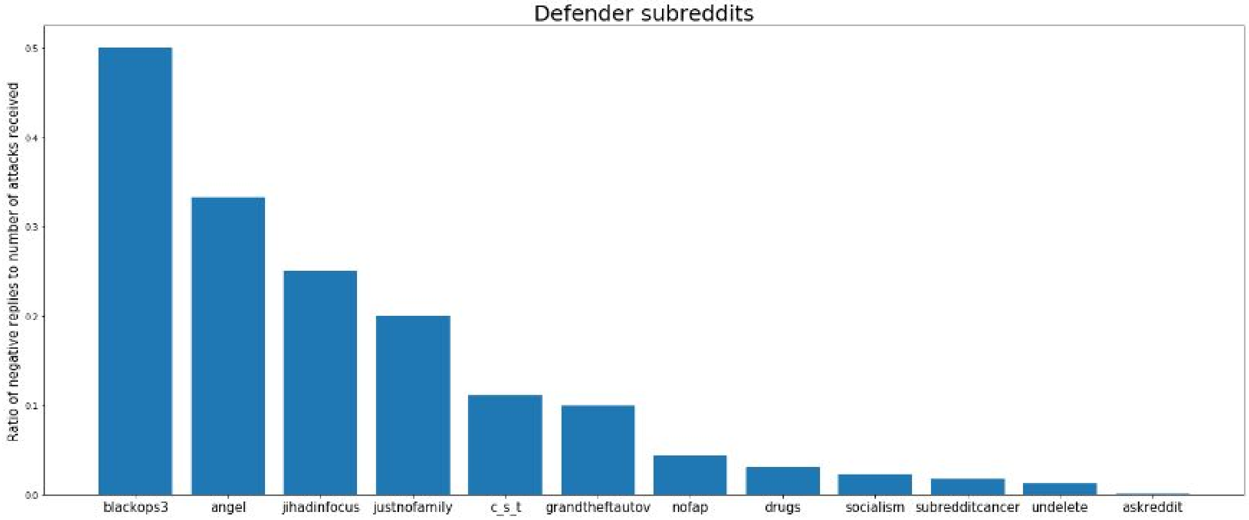
Insights and Visualization

Disclaimer: For better quality images, please refer to the python notebook files provided along with this report.

1. **Top 10 defender subreddits**

   Since we don't have data in the scale that authors of Community Interaction and Conflict on the Web. World Wide Web Conference, 2018 had, we thought of coming up with our own statistics from the kind of data we have to find defender subreddits.

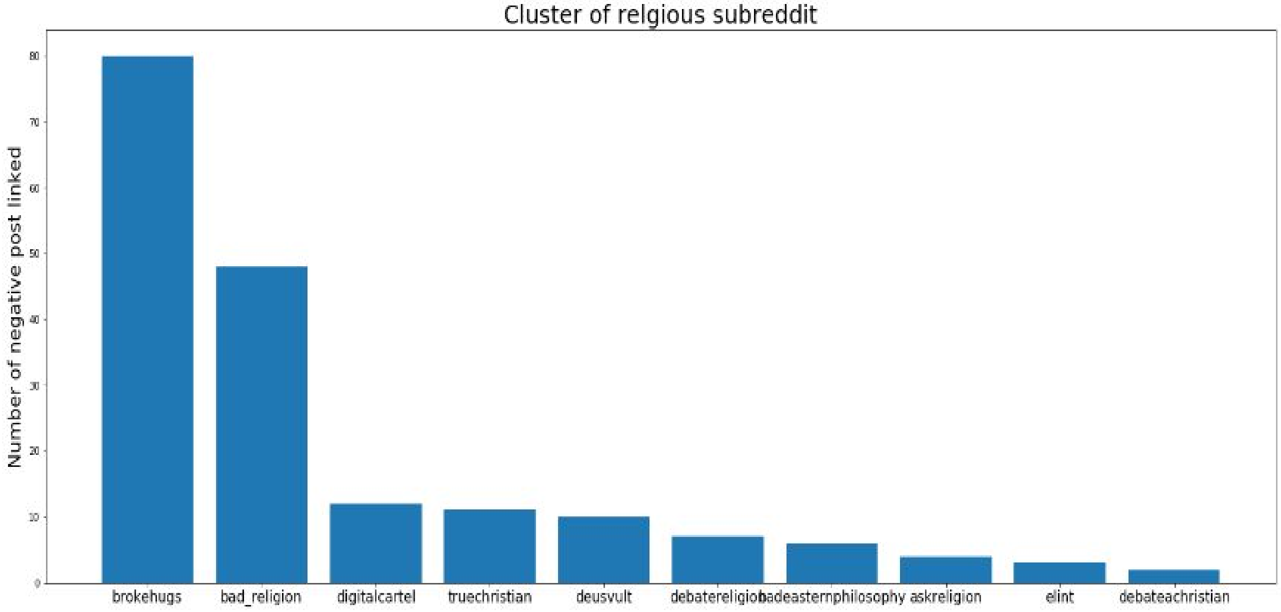
   We basically try to count the number of negative replies a subreddit makes within 24 hours of receiving an attack from other subreddit (assumption: defence is when A attacks B within 24hours when it receives an attack from B). In other words, for every negative cross-link (L1) a subreddit A receives from subreddit B, we try to find a negative edge (L2) from subreddit A to B such that L2.timestamp - L1.timestamp is less than 24hrs. We find the total number of such loops for every subreddit (negative_reply_count) and then we calculate the ratio(d) = negative_reply_count / total number of negative cross-link it has received. Thus, the subreddit with highest value for d (defence value) value is assumed as defenders. Some of the defenders are blackops3, angel, jihadinfocus,etc



This is just a heuristic and we believe we need more detailed data to accurately determine defender subreddits.

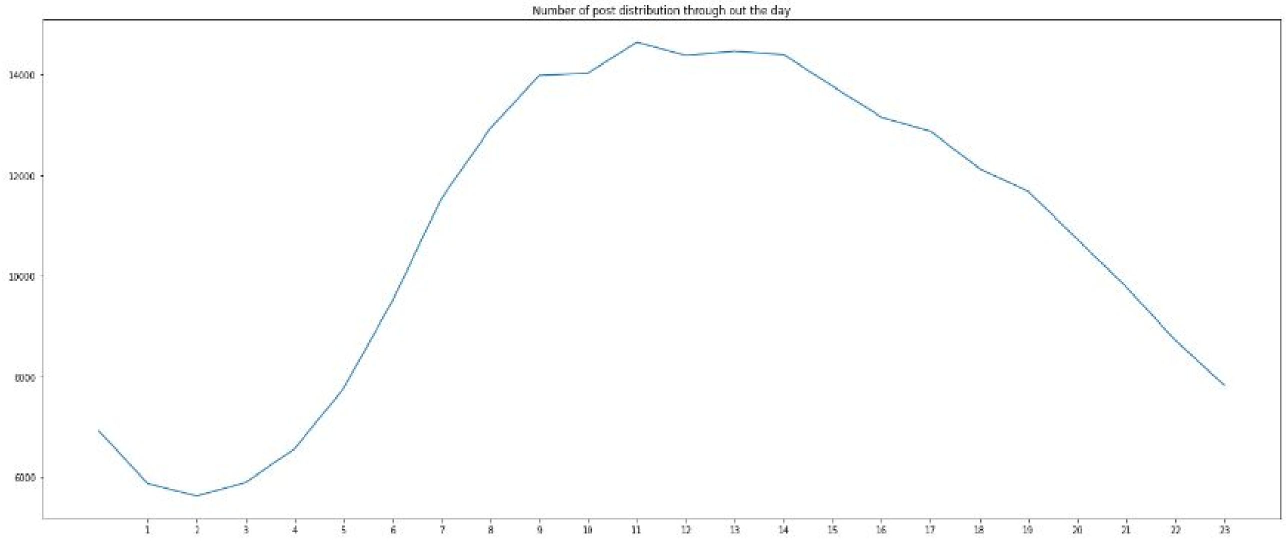
2. **Religious Subreddits**

   The 81th text property of the post estimates the religious context of the post. We have used this property to cluster religious subreddits.



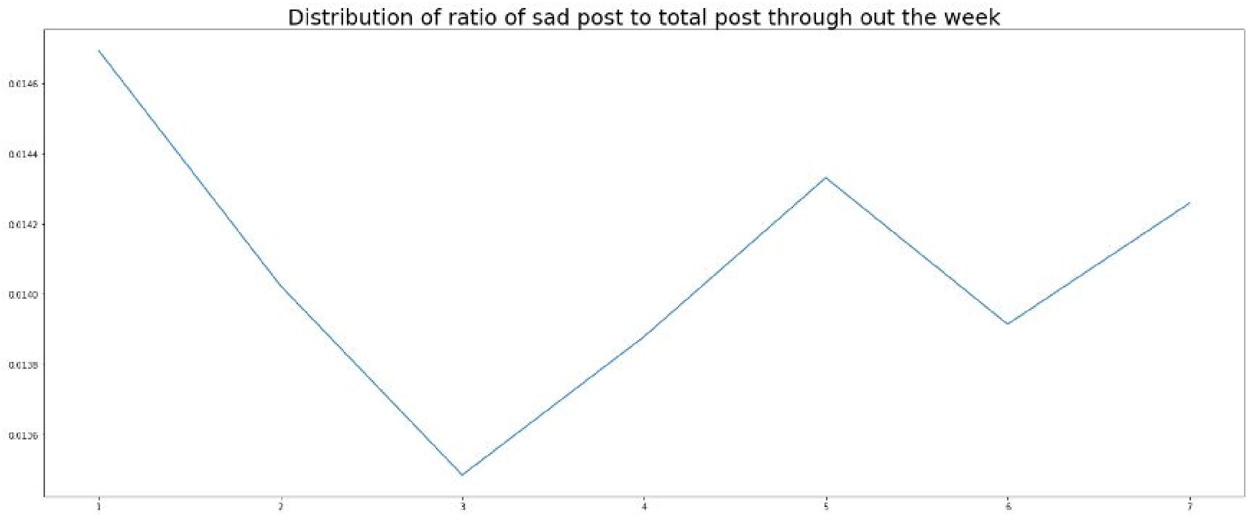
   The y-axis shows the number of post cross-linked from a subreddit which have a value greater or equal to 0.03 for LIWC_relig property.

3. **Post distribution in reddit over the day**



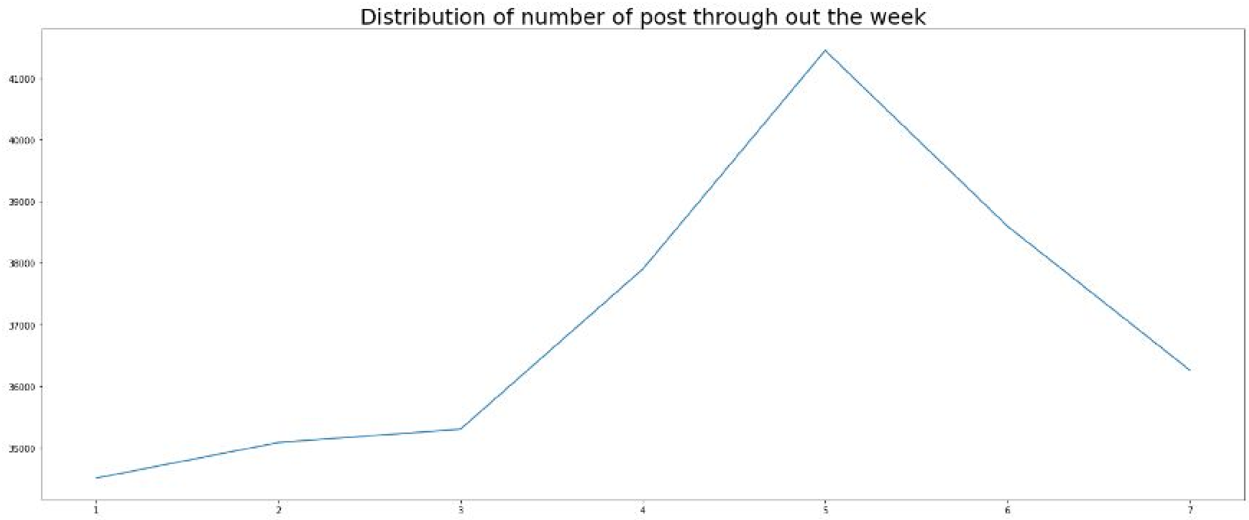
   It can be seen that reddit gets busiest from 9 am to 4pm and it's absurd to see that the aforementioned time range falls during office/school hours.

4. **Distribution of sad post over the week.**



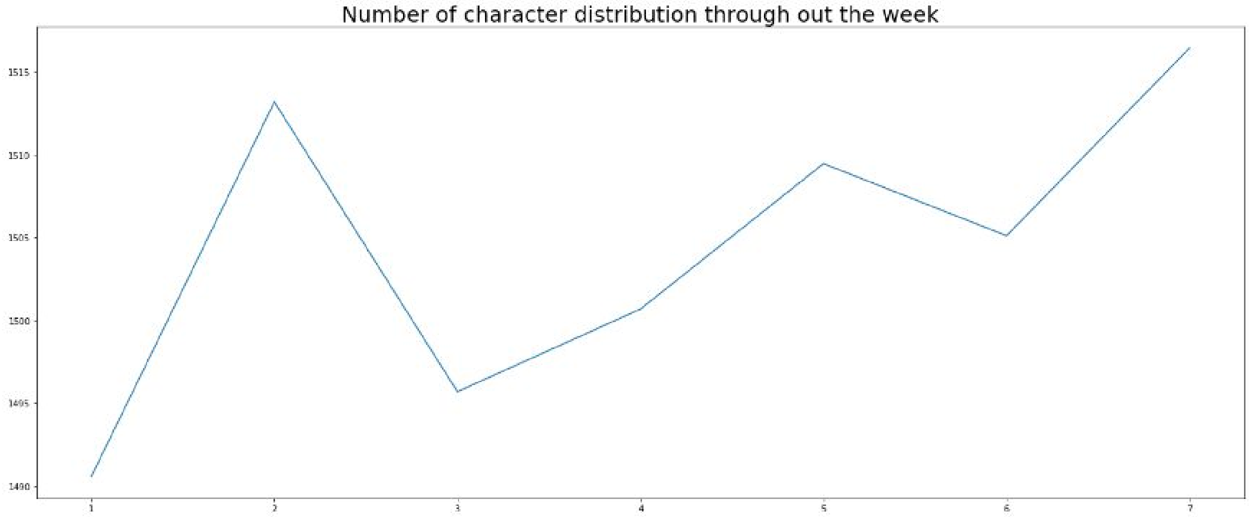
The y-axis shows the ratio of number of sad post and total number of posts. The day with the most number of posts would have high weightage that is why such ratio is needed to be calculated instead of directly calculating number of sad posts. Monday has the highest value for sad post ratio (which is obvious because people are back to work/school) while saturday and wednesday has least value for sad post ratio. A post is considered sad if LIWC_sad has value greater than equal to 0.02. **In such insights**. **we have an assumption that trend of cross-link implies similar behaviour within the group.** We have to make such an assumption because the data we have shows the behaviour of cross-links not within the subreddits.

5. **Distribution of posts throughout the week**



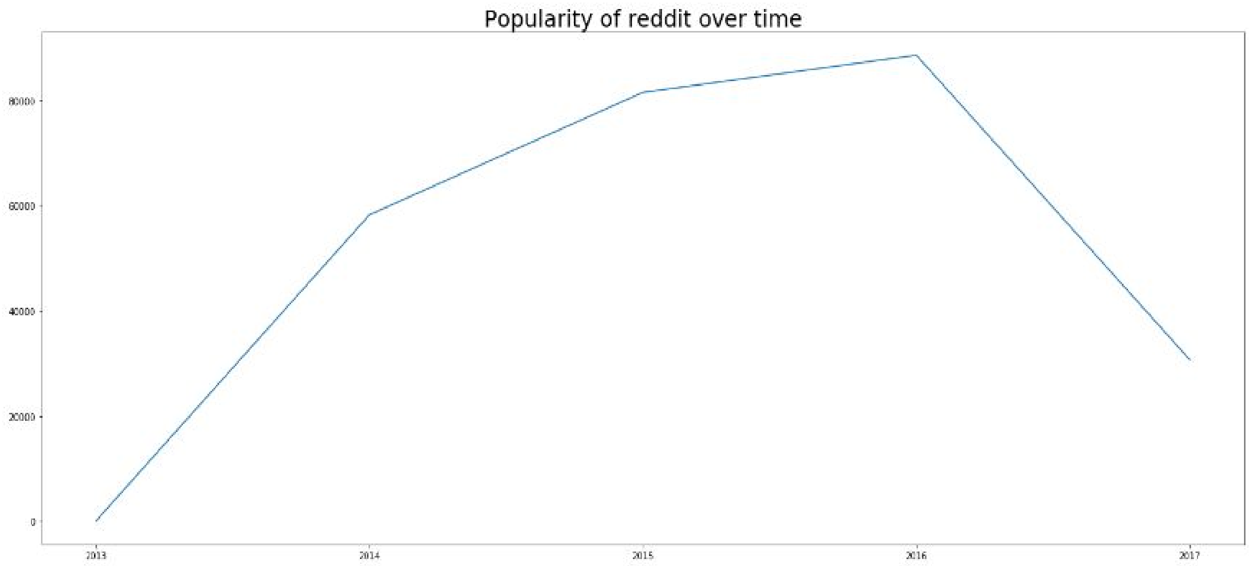
Friday has the most number of cross-links, and this is because people ,maybe, have more free time on friday to engage in conversation.

6. **Distribution of post length throughout the week**



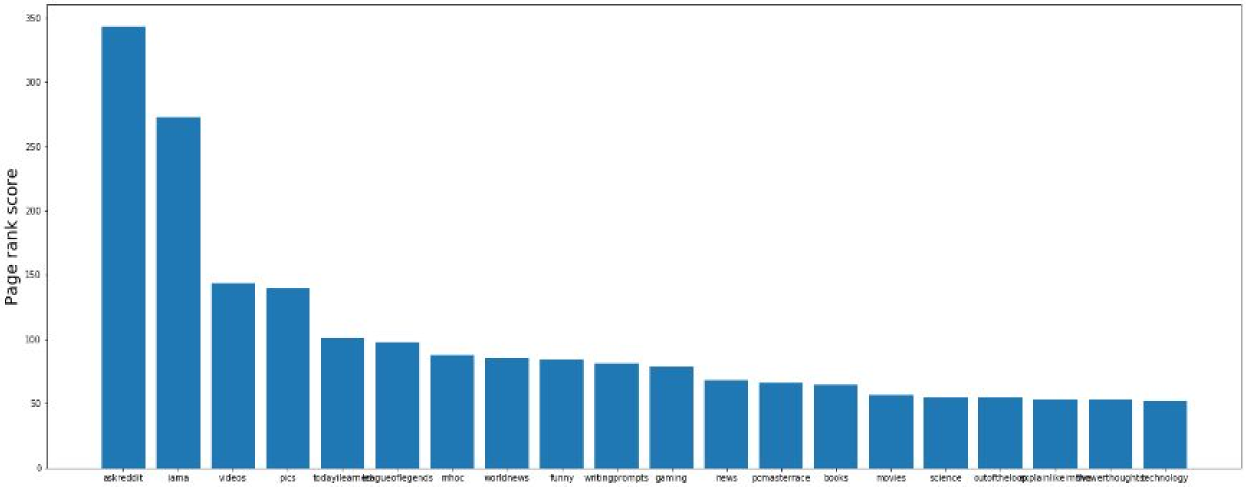
It can be seen that people on sunday use lengthy post compared to other days. This is maybe because people have more time on weekends to write long posts.

7. **Popularity of Reddit over the years**



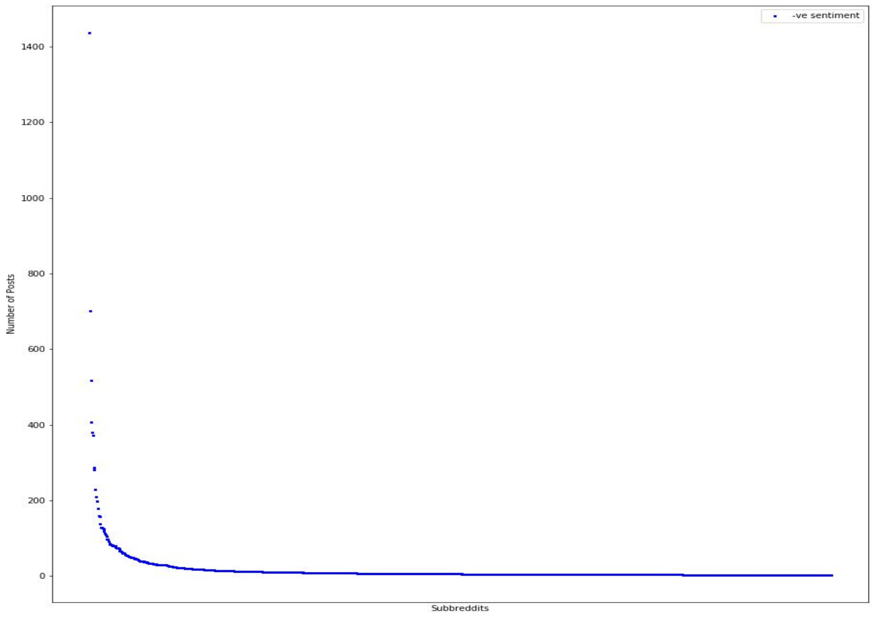
It can be seen that popularity of reddit is highest in 2016. Events like **brexit, trump presidency** might have led the people to engage more in social networks. Also, the rise of meme culture would have led the increase in the popularity of reddit. It is absurd that the number of cross-links are less in 2017, this maybe because of incomplete dataset or maybe because of reddit’s stricter policy to cross-link? (just a hypothesis)

8. Top 10 popular of subreddits



For finding out the popularity of the subreddits, we have calculated the pagerank score for each subreddits. It can be seen that askreddit, Iama, videos are the most popular subreddits.

9. Behavior of subreddits for positive and negative sentiment cross-links



1) Above plot shows number of posts with -ve sentiment against the subreddits.

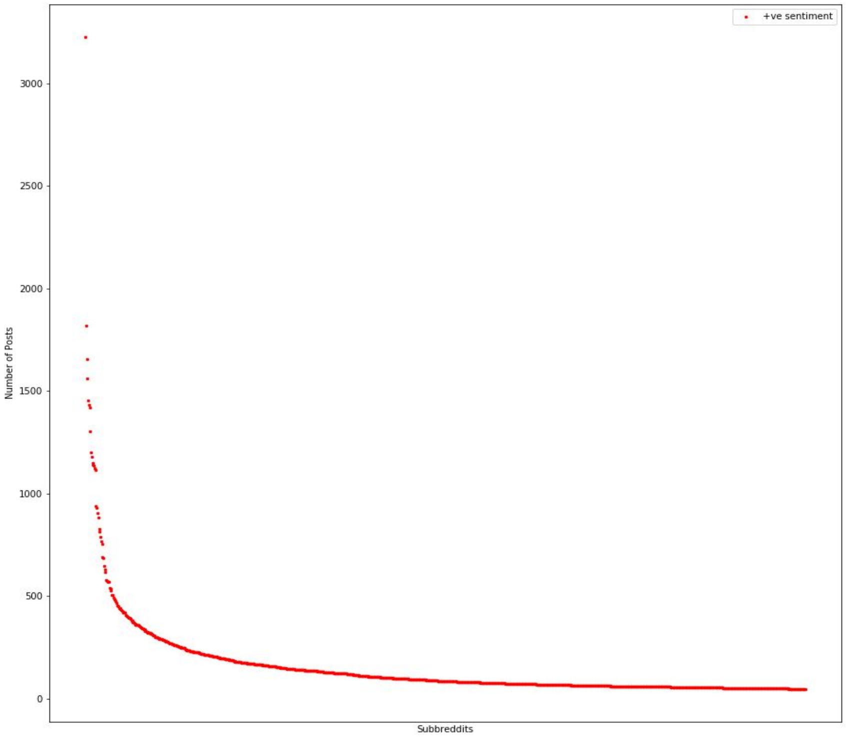
2) The graph formed is very steep which means people don't reply to negative posts which results in **low engagement**.

3) By reply we mean that consider a scenario where Subreddit A writes negative posts targeting Subreddit B and Subreddit B **did not reply**. Meaning A is the source

Subreddit and B is target Subreddit, when it comes to reply B should source and A the target.

4) This scenario is more with -ve sentiments that’s why the slope of the graph is less.

5) Also, highest no. of -ve posts are above 1400.



1) Above plot shows number of posts with +ve sentiment against Subreddits.

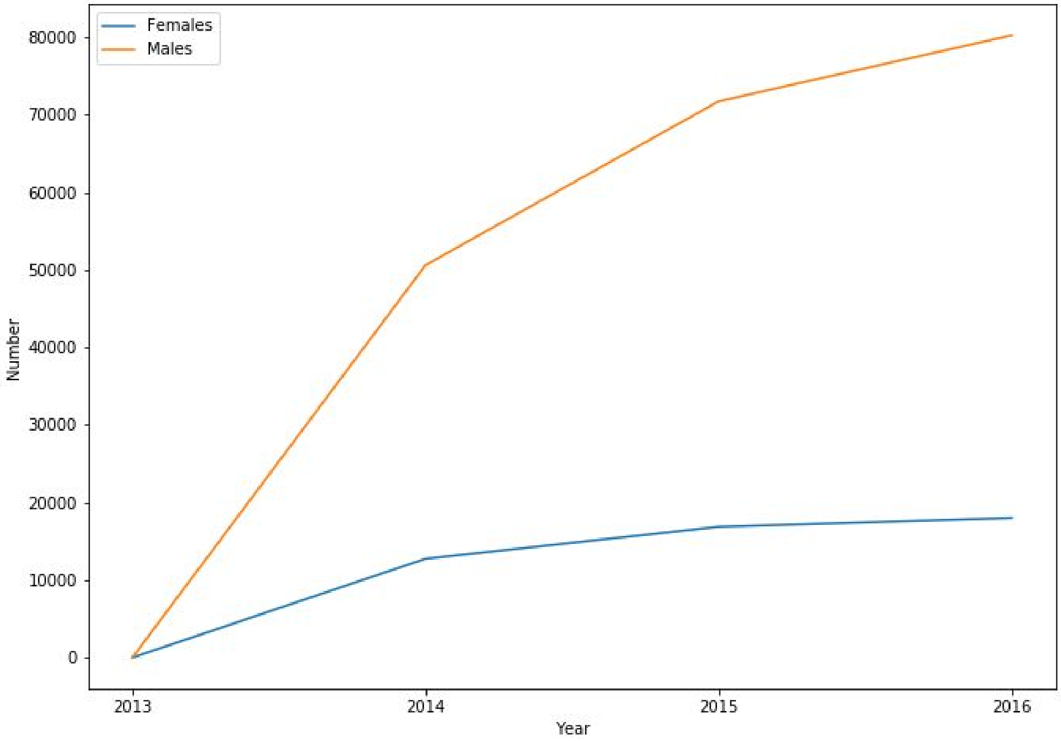
2) The graph formed is not that steep when compared with above graph which means people reply to positive posts resulting in **high engagement**.

3) By reply we mean that consider a scenario where Subreddit A writes positive posts targeting Subreddit B and Subreddit B **did reply**. Meaning A is the source Subreddit and B is target Subreddit, when it comes to reply B should source and A the target.

4) This scenario is more with +ve sentiments that’s why the slope of the graph is more compared to the plot with -ve sentiments.

5) Also, highest number of +ve posts is above 3000.

10. Likely Number of males and females in Subreddits



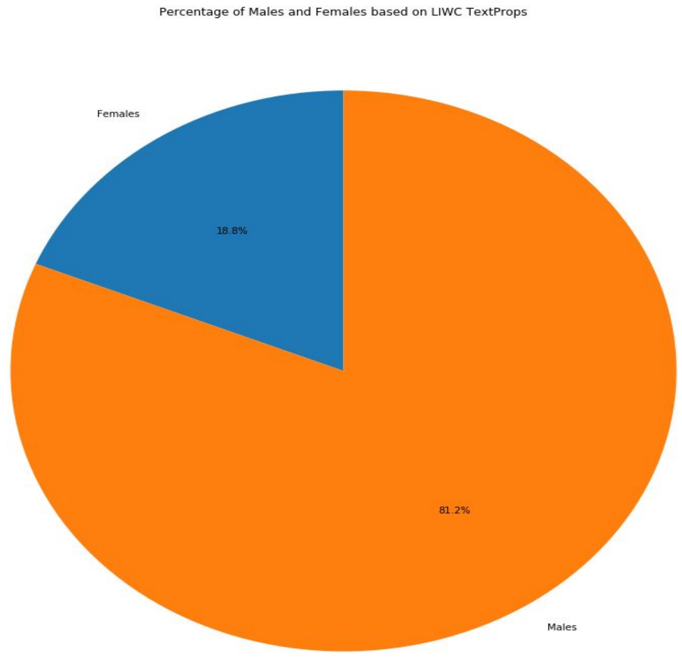
1) Based on LIWC TexProps we tried to classify that whether a post was written by a male or a female user. The LIWC properties that we considered for **females were 23 27 28 29 and males were 38 31 25 26**.

2) Females tend to use more pronouns, 2nd and 3rd person narrative whereas males use more articles, prepositions and 1st person narrative in their speech or while writing.

3) Males and females have different speech styles as talked by **Yla R. Tausczik and James W. Pennebaker** in **The Psychological Meaning of Words: LIWC and Computerized Text** [1][3].

4) Additional documents “Male and female speech styles.pdf” and “LIWC2007LanguageManual.pdf” we will be attaching along with the assignment for more reference.

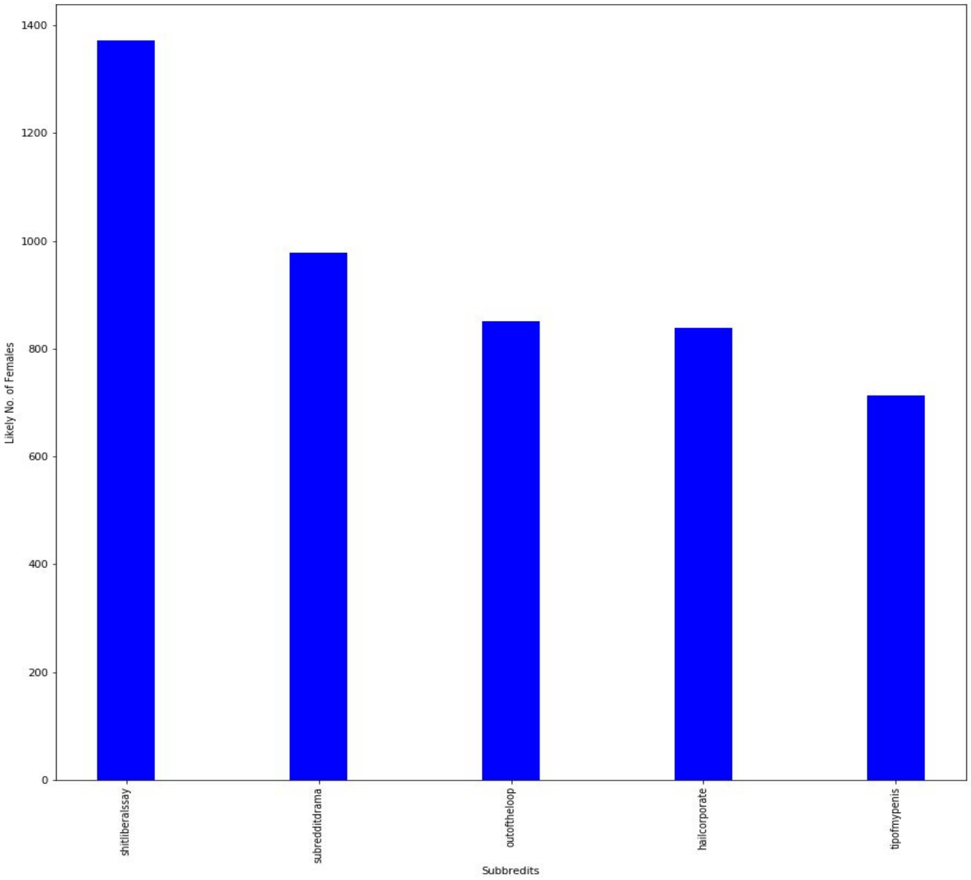
5) From this plot we can see that number of males and females have increased over the years and is still growing. Scaling the data and including data for years 2017,2018 and 2019 would also show similar trends.



1. Above pie chart shows that reddit has **81.2% male users and 18.8% female** users according to the dataset provided which is quite close to the actual numbers which are **71% males and 29% females**.[2].

2. Scaling the data and including more Subreddits and data from more years, its expected the we will get **similar results**.

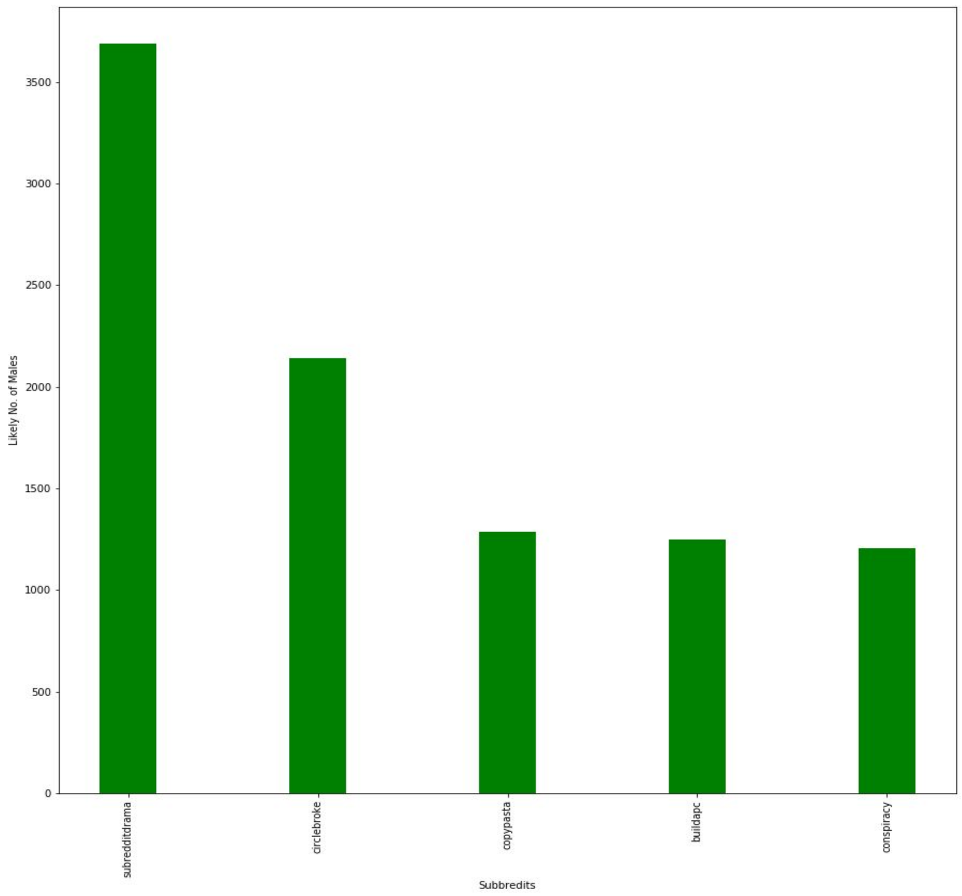
3. Also, it is our attempt to **create a very basic heuristic model** which classifies whether the user is a male or a female based on TextProps provided [1].



1. Above graph shows top 5 subreddits with highest number of females.

2. Shitliberalssay has the highest number of female users according to our prediction.

3. Highest number of females is around 1300 according to our model.

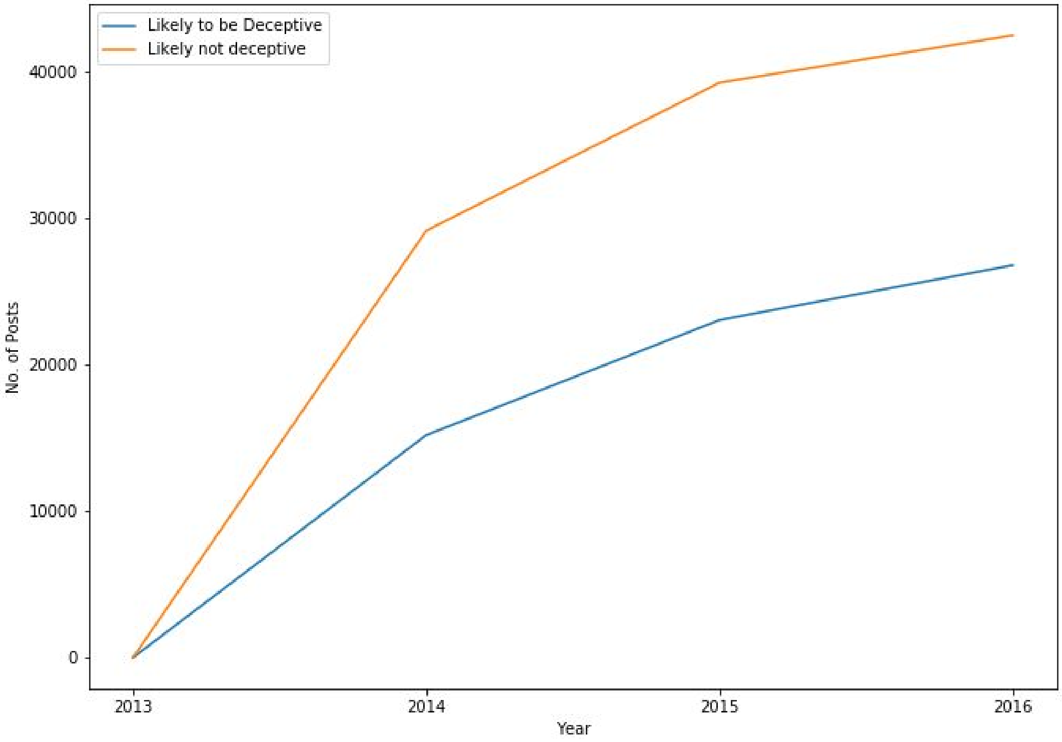


1. This graph shows top 5 subreddits with highest number of males.

2. Subredditdrama has the highest number of males.

3. Highest number of males is above 3500 according to our model.

11. If the Post is Deceptive or Not



1) Based on LIWC TexProps we tried to classify that whether the post is deceptive or honest. By deceptive we mean that if the post is misleading. The LIWC properties that we considered to classify the post as **Deceptive are 73 50 62 25.**

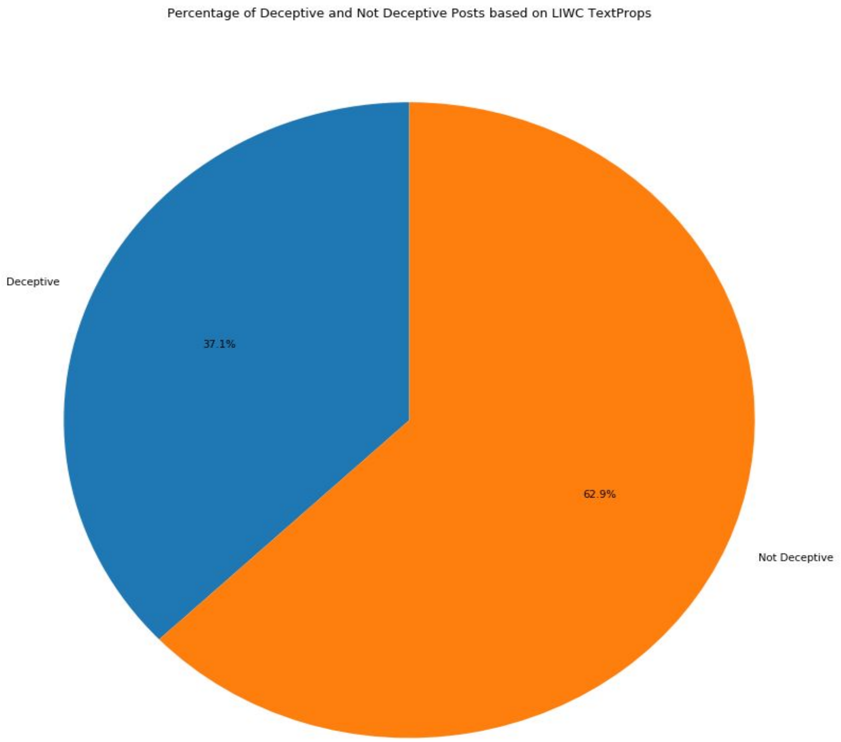
2) Deceptive texts have more motion words, negative emotions, exclusion words and use 1st person narrative.

3) Deceptive texts have some properties as talked by **Yla R. Tausczik and James W. Pennebaker** in **The Psychological Meaning of Words: LIWC and Computerized Text** [1][3].

4) Additional documents “LIWC2007LanguageManual.pdf” we will be attaching along with the assignment for more reference.

5) From this plot we can see that number deceptive/misleading posts have increased over the years and is still growing. Misleading information is a major problem of today’s information Era and models and algorithm are being designed to recognize

them, for example Facebook implemented AI Algorithm to automatically detect and delete Fake news and misleading information.

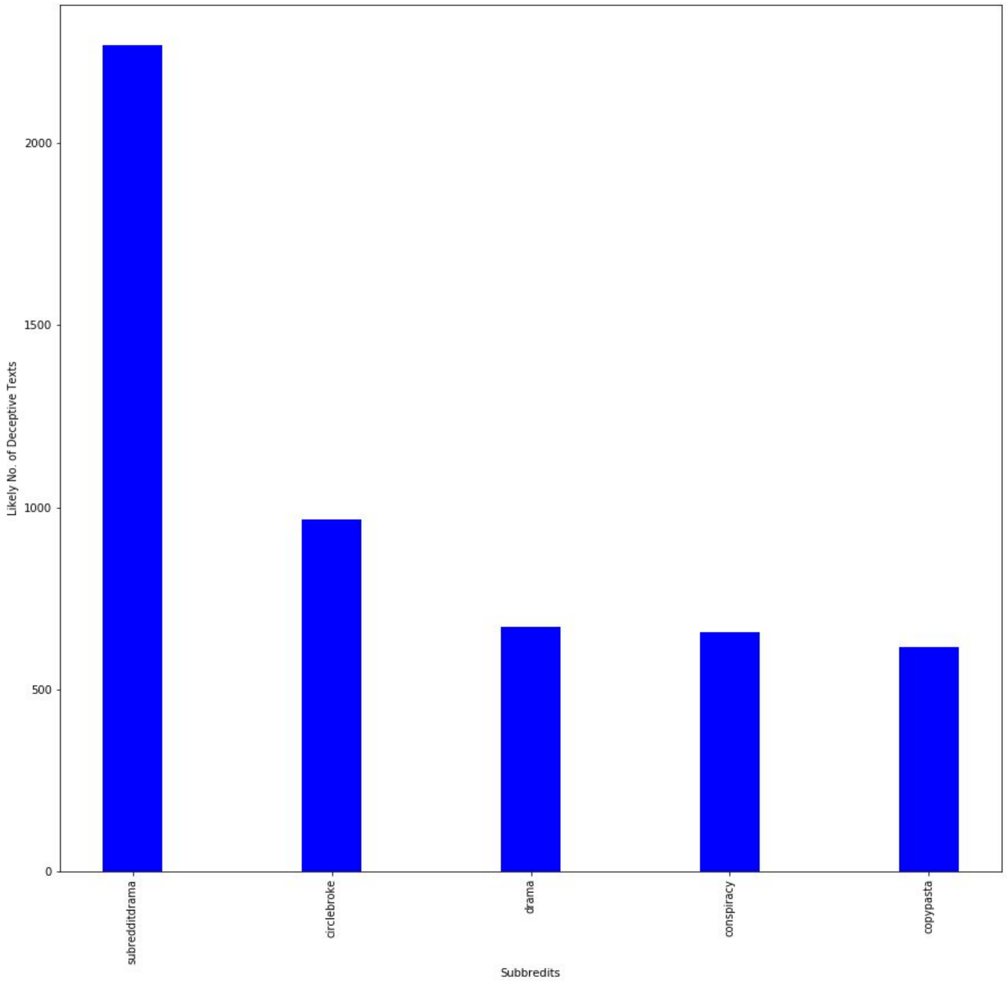


1) Above plot depicts that 37.1% of the posts are deceptive/misleading and 62.9% are honest.

2) It is our attempt to **create a very basic heuristic model** which classifies whether the post is deceptive or not based on TextProps provided [1].

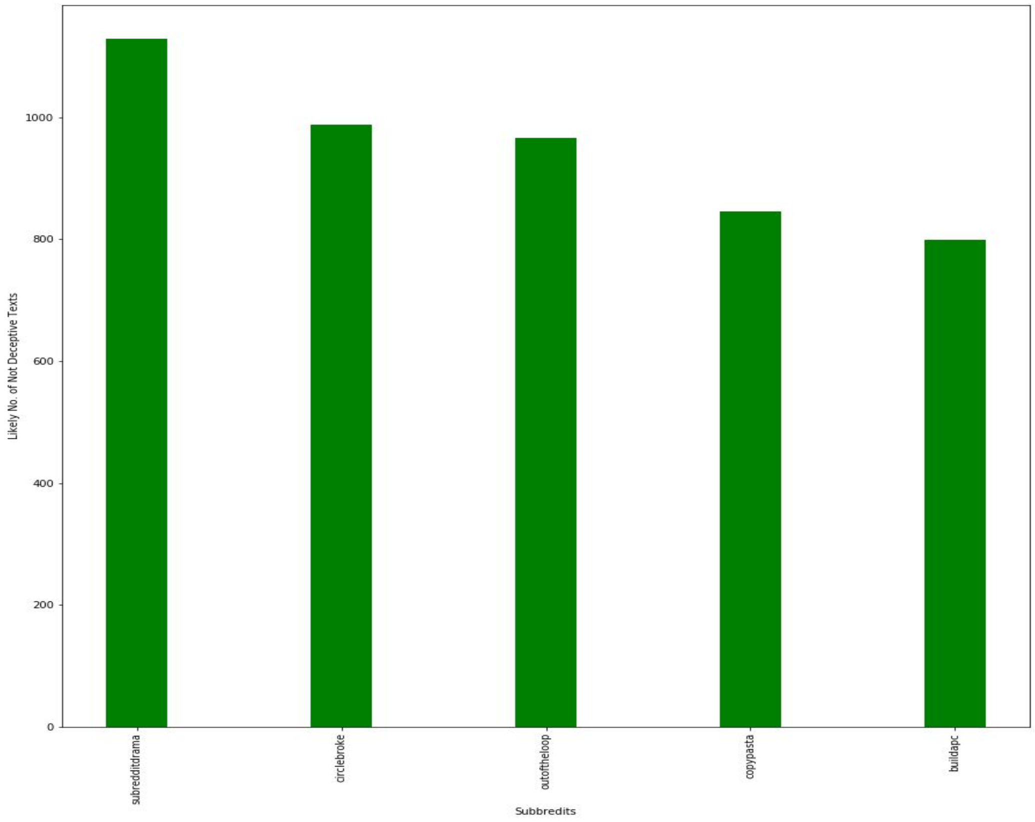
3) Such advanced models are necessary to filter out fake news and misleading posts so that social media and other websites are a safe and informative experience.

4) Fake news has been the cause of lot problems all around the world and its mostly circulated through social media.



1) This plot shows that subredditdrama has most deceptive posts. This subreddit is about people writing posts about what’s going on in other subreddits. Subredditdrama has most -ve sentiment posts because people there tend enjoy the fights going around, so it’s natural for this subreddit to have highest number of deceptive posts.

2) Also, conspiracy subreddit is at 4th number which seems fair because in this subreddit people post conspiracy theories which can be untrue and misleading.



1) Subbreditdrama has the highest number of non-deceptive posts but these are less the deceptive posts as subredditdrama has the most deceptive posts too.

2) Buildapc subreddit is about to building custom computers which are more facts that's why Buildapc has high non-deceptive posts.

12. Cluster coefficient and number of triangles

For determining the compactness of the graph, we calculated the average cluster coefficient using neo4j's implementation of algorithm, and it was found to be 0.09633692818747569, which is quite low for a popular social media like reddit. Also, the total number of triangles in the graph is 406391.

Adherence to the Rubric

1. We have successfully extracted interesting temporal and behavioral insights from the data by modelling it in the form of graph. We have successfully utilised the text property of the post to extract behavioral insights like religious nature of subreddits, gender dominance, defensive and deceptive nature of post. We have shown temporal insights like the time and day of the week when the subreddit gets busiest, popularity of subreddit and also, the day when most number of sad post are cross-linked. In total, we team of 2 people have managed to extract 12 insights.

2. Our approach to graph modelling has been explained in ETL section in detail. Since, we are interested in temporal aspect of the insight, we created separate nodes for hour, day and year which helped to make the queries performant. Also, we have denormalized the structure (redundant edge :target between subreddit nodes) which reduced the number of edge traversal and also, the redundant edge helped in calculation of pagerank and cluster coefficient more efficiently. Thus, our design choice helped us perform OLAP operations more efficiently.

3. Use of indexes on nodes and relationship made the queries performant. The use of query planner to detect eager loading helped us to load full graph into database and made our approach more scalable. This helped us to **load whole graph within 3 minutes**. Also, we have shown how we restructured our graph to speed up queries for extracting temporal insights. Most of the pattern matching used in the Cypher query involves breadth first search traversal accompanied with indexes. Also, we built our own simple heuristic to calculate complex behavioral insight in an easy manner.

4. A) We have taken references from **The Psychological Meaning of Words: LIWC and Computerized Text** by **Yla R. Tausczik and James W. Pennebaker.**
    1. We can tell a lot of things from a person's speech and writing style, for example maturity level, sex, psychological processes, status etc.
    2. We have done few visualizations using this information.
    3. Whether the person posting is male or female, whether the post is deceptive or not.
    4. These visualizations take reference from the research paper mentioned above. The author clearly explains how the above information from a

person’s text or speech can help us know a lot of things, such techniques are also used in lie-detection test too.

B) Also we have used Indexing in this assignment to reduce the time taken for the data to load in Neo4j and create graph from it. Neo4j uses Hash Maps for indexing. The research paper “**Nanosecond Indexing of Graph Data With Hash Maps and VLists**” talks about indexing.

1. In Neo4j we can create index over the properties on any node that has been given a label. Once these indexes are created Neo4j will manage and update them whenever the database is changed.
2. When Neo4j creates an index, it creates a redundant copy of the data in the database. Therefore, using an index will result in more disk space being utilized, plus **slower writes to the disk**, which is also talked about in the research paper. Generally, it's a good idea to create an index when we know there's going to be a lot of data on certain nodes.In our scenario, there won’t be any overhead caused by indexes because nodes are not modified as we are performing only OLAP operations. Also, if the queries are taking too long to return, adding an index may help.

References

[1] The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods.
By Yla R. Tausczik and James W. Pennebaker.
DOI: 10.1177/0261927X09351676

[2] https://www.techjunkie.com/demographics-reddit/#Age_and_Gender

[3] https://www.liwc.net/LIWC2007LanguageManual.pdf