UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMA
ZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN
ICT For Internet and Multimedia

"Semantic network analysis of Twitter data and their psycho-social implications"

Relatore: Prof. Tomaso Erseghe

Laureanda: Lejla Džanko

Correlatore: Prof. Caterina Suitner

ANNO ACCADEMICO 2021– 2022

Data di laurea 28.02.2022

List of images 4

List of tables 5

1. Introduction 6

2. Main concepts from network science [1] 7

3. Comparing the role of men within prolife and prochoice community in discussion about abortion on Twitter 8

3.1 Project introduction 8
3.2 Software tools 9
3.3 Data collection and preliminary analysis 9
3.4 Cleaning the data 16
3.5 Network analysis 17
3.5.1 Methods 17
3.5.2 Results 18
3.5.2.1 Word-hashtag bipartite network 18
3.5.2.2 Word-topic bipartite network 21
3.5.2.3 Hashtag-topic bipartite network 24
3.5.2.4 Word+hashtag-topic bipartite network 26
3.6 Conclusion 30

4. Analyzing political discourse before and after election using Tweets posted by the candidates for the 2020 US Elections 31

4.1 Project introduction and goals 31
4.2 Software tools 31
4.3 Data collection and preliminary analysis 32
4.3.1 Elected members 33
4.3.2 Candidates who lost the election 34
4.3.3 Collecting Tweets 35
4.4 Data cleaning 35
4.5 Network analysis 38
4.5.1 Methods 38
4.5.2 Results 39
4.5.2.1 Chosen members/democrats/before 40
4.5.2.2 Chosen members/democrat/after 41
4.5.2.3 Chosen members/republican/before 42
4.5.2.4 Chosen members/republican/after 43
4.5.2.5 Losers/democrat/before 44
4.5.2.6 Losers/democrat/after 45

4.5.2.7 Losers/republican/before 46

4.5.2.8 Losers/republican/after 47

4.5.2.9 Discussion 48

4.6 Conclusion 52

5. Summary and conclusion 53

6. References 56

Code appendix 57

1.Authorization header creation 57

2. Twitter API data collection 57

2.1 Counts endpoint 57

2.2 Search tweets endpoint 58

2.3 Search users endpoint 60

2.3.1 Winners 60

2.3.2 Losers 61

2.4 Users timeline endpoint 62

3. Wikipedia scraping 63

3.1 Winners 64

3.2 Losers 65

4. Text cleaning 67

5. Create network graph 68

5.1 Color the nodes 69

5.1.1 Topic network 69

5.1.2 Hashtag network 69

5.2 Create word projection 70

6. Community correlation 70

7. Similarity of topic to network 73

List of images

Image 3.1 : distribution of tweets over time; a) prolife (our dataset), b) prochoice (our dataset), c) prolife (overall), d) prochoice (overall)..12
Image 3.2: word ranking in prolife and prochoice (top 20 words ranked by frequency).............13
Image 3.3: frequency of male-related keywords in the prolife and prochoice datasets…………14
Image 3.4 : hashtag ranking in prolife and prochoice (top 20 words ranked by frequency).........14
Image 3.5 : frequency of male-related keywords (as hashtags) in the prolife and prochoice hashtag datasets……………………………………………………………………..15
Image 3.6: percentage of tweet types in a) prolife b) prochoice dataset…………………………16
Image 3.7: Number of retweets for most retweeted tweets in a) prolife b) prochoice dataset….16
Image 3.8: prolife bipartite word-hashtag network………………………………………………19
Image 3.9: prochoice bipartite word-hashtag network..…………………………………………19
Image 3.10: prolife bipartite word-topic network………………………………………………..22
Image 3.11: prochoice bipartite word-topic network……………………………………………22
Image 3.12: prolife bipartite hashtag-topic network……………………………………………..25
Image 3.13: prochoice bipartite hashtag-topic network…………………………………………25
Image 3.14: prolife bipartite word+hashtag-topic network……………………………………...27
Image 3.15: prochoice bipartite word+hashtag-topic network…………………………………..27
Image 4.1: twitter profile data of the elected members………………………………………….33
Image 4.2: twitter profile data of the candidates who lost the election………………………….34
Image 4.3: percentage of tweet types in a) chosen members b) candidates who lost the election dataset……………………………………………………………………………………………36
Image 4.4: tweet count across different time period, party affiliation within winner and loser datasets…………………………………………………………………………………………..37
Image 4.5: communities in the chosen_members_democrat_before network…………………..40
Image 4.6: communities in the chosen_members_democrat_after network…………………….41
Image 4.7: communities in the chosen_members_republican_before network…………………42
Image 4.8: communities in the chosen_members_republican_after network……………………43
Image 4.9: communities in the losers_democrat_before network……………………………….44
Image 4.10: communities in the losers_democrat_after network………………………………..45
Image 4.11: communities in the losers_republican_before network…………………………….46
Image 4.12: communities in the losers_republican_after network………………………………47
Image 4.13: community correlation matrix……………………………………………………...48
Image 4.14: spatial configuration of the nodes in a) two and b) three dimensions……………..50
Image 4.15: spatial configuration of the nodes with labels……………………………………...51

List of tables

Table 3.1: overview of tweet count across different datasets……………………………………11
Table 3.2: overview of the word-hashtag networks……………………………………………...18
Table 3.3: overview of the word-topic networks………………………………………………...21
Table 3.4: overview of the hashtag-topic networks……………………………………………...24
Table 3.5: overview of the word+hashtag-topic networks……………………………………….26
Table 4.1: overview of the eight networks in terms of size (number of nodes and edges) and number of communities……………………………………………………………………...39
Table 4.2: overview of the communities in the chosen_members_democrat_before network…..40
Table 4.3: overview of the communities in the chosen_members_democrat_after network…….41
Table 4.4: overview of the communities in the chosen_members_republican_before network…42
Table 4.5: overview of the communities in the chosen_members_republican_after network…...43
Table 4.6: overview of the communities in the loser_democrat_before network………………..44
Table 4.7: overview of the communities in the loser_democrat_after network………………….45
Table 4.8: overview of the communities in the loser_republican_before network………………46
Table 4.9: overview of the communities in the loser_republican_after network………………...47

Abstract

To study the language of social networks and its psycho-social implications there is a need for an interdisciplinary approach that combines scientists from the fields of network science, psychology and linguistics. This thesis is a product of the author's work undertaken in two such interdisciplinary projects, Comparing the role of men within prolife and prochoice community in discussion about abortion on Twitter and Analyzing political discourse before and after election using Tweets posted by the candidates for the 2020 US Elections. Both projects used tweets in English language from various users collected over a predefined time span. Twitter data offers various possibilities for interpretation within the context of network science. The focus of this thesis was to study a special kind of networks, the semantic networks, built from the tweet text, hashtags and metadata. More specifically, we detected meaningful communities based on the topics in order to study the language used within the context of those topics. Although in both projects tweets are used to study language on social media and its psycho-social implications, the main goal differs and so does the methodology applied to community detection.

1. Introduction

One of the main topics studied in the field of network science is, naturally, analyzing social networks and their dynamics. Network science analysis helps us understand how ideas and information spread through networks, how robust networks are to outside influences and how the nodes in a network organize into communities. Online social networks are also of great interest to researchers. The data is, for the most part, readily available and large in volume, and the sample is well varied over different social categories. Finally, we can observe not only social networks in a traditional sense, as networks of users that interact, but also the language that is used there in order to learn something that is innate to a network. To study the language of social networks and its psycho-social implications there is a need for an interdisciplinary approach that combines scientists from the fields of network science, psychology and linguistics. This thesis is a product of the author's work undertaken in two such interdisciplinary projects. Both projects used Tweets in English language from various users collected over a predefined time span. Since there are significant differences in the projects, they are going to be shown separately.

First project that is going to be shown tries to demonstrate how the role of male partners in abortion is perceived in prolife and prochoice community. The data collection was keyword-based as the Tweets were not observed on a user level, rather, we tried to compare the discourse in two opposing communities where belonging to a community was based on the use of “prochoice” or “prolife” in the Tweet text.

The second project studies the language used in Tweets posted by the candidates for the Senate and the Congress in the 2020 US elections. Data collection was username-based and done in the nine months preceding and nine months following the elections. The main aim of the project is to observe the level of agency through time, both on the user and group level, and see if it rises significantly in time before and just after the election. Here we are going to focus on comparing the discourse before and after election, in elected members and those who lost the election and the republican versus the democrat discourse and note the differences.

The results will consist of statistical data, visualizations and interpretation based on the network science theory, supplemented by social psychology and linguistics.

2. Main concepts from network science [1]

Network science studies the properties of complex networks, such as computer, scientific, biological, semantic and social networks. There are different analyses that can be done on the network; we decided to focus on community detection and PageRank of the nodes.

A **network** is a graph represented by its nodes and edges (connections between the nodes). A network can be **directed** or **undirected**. In an undirected graph the edges indicate a two-way relationship (you can traverse the graph from source to target node and vice versa) while directed graphs have edges that indicate a one-way relationship (you can only traverse the graph from the source to the target network. In our work we relied on the concept of undirected bipartite networks and their projection. A **bipartite network** is a graph whose vertices can be divided into two disjoint and independent sets U and V, such that every edge connects a node in U to one in V. A **projection of a bipartite network** on set U consists of the nodes from U connected with a (possibly weighted) edge if they share a common connection in V.

An **edge** can be binary (0 or 1) or weighted (assigned a real-number weight). In this work we used weighted edges.

A **node** in the network can be characterized by different attributes. Most important ones are the node's degree, centrality and community affiliation. Degree is a number of connections to other nodes that a node has. **Centralities** measure a node's "importance" in a network. The most studied centralities are closeness, prestige or eigenvector and betweenness. We will focus on a special eigenvector centrality called **PageRank [2]**. It was designed by Google founders for ranking web content, using hyperlinks between pages as a measure of importance but is widely applied to all kinds of networks.

Communities are sets of nodes that are similar to each other. To detect communities we use the **Louvain method [3]**. It evaluates how much more densely connected the nodes within a community are, compared to how connected they would be in a random network.

3. Comparing the role of men within prolife and prochoice community in discussion about abortion on Twitter

3.1 Project introduction

While abortion remains a hot topic, little is known about the how male partners enter in the discussion as targets. The goal of the project is to analyze discussion about the role men have in the volountary abortion decison making by observing language used to discuss this topic on Twitter. We look at the prochoice and the prolife group separately and compare their discourse. We collected 26411 Tweets (17505 prochoice, 8906 prolife) published in 2019 and that contain words: men, father, dad, husband, boyfriend and either prochoice or prolife. We apply standard tools of network science to create semantic networks of words, hashtags and topics and detect communities to understand which arguments are used to talk about these topics. We compare the results within the prochoice and prolife dataset and across different types of networks. The identification of communities in the network is obtained by applying Louvain modularity. The importance of a community is interpreted as the sum of the PageRank centrality scores of the words belonging to that specific community. We noted that prolife Tweets are higher in volume and more evenly distributed through time, while the prochoice Tweets are centered around important events (such as abortion ban in certain US states). In both datasets the word men was used to denote lawmakers and politicians rather than male partners. In neither dataset did the discussion about the role of male partners achieve significant centrality. They also differ in treatment of keywords we used, with “father” being used to refer to God in the prolife dataset. We conclude that both groups include male partners in the discussion only if their opinions are in line with the group sentiment.

3.2 Software tools

For data collection, cleaning and network building we used Python 3.7.12. The code was written, tested and executed inside the Google Colab Notebook environment, which is an online Jupyter notebook that offers online computational resources and is especially well suited to data analysis since the most used data analysis libraries are already installed in the environment.

To collect tweets we communicated with the official Twitter API, version 2.0, using the Python requests library. We used an account with Academic research access, which allows us to fetch up to 10 million tweets per month, access to all endpoints, query parameters and operators as well as the entire historical archive of Twitter. We collected the tweets using `tweets/search/all` endpoint and validated tweet count using the `tweets/counts/all` endpoint.

To clean the text we used nltk (natural language toolkit) and re (regex) libraries. For data storing and manipulation we relied on the pandas library.

To create network graphs from our data we used networkx, a network science library in Python.

For statistical visualization, we relied on matplotlib. To visualize network graphs, calculate PageRank and perform community detection we used Gephi, an open source software.

3.3 Data collection and preliminary analysis

To capture the discussion on male role in abortion on Twitter, we performed a search by using male nouns (men, father, dad, husband or boyfriend) as keywords, along with prolife or prochoice as an indicator of the tweet's author taking a prolife or prochoice stance on the abortion. We decided not to search by hashtags since it is not likely that our keywords are used as hashtags as often as they are used as words in a sentence. Moreover, search by keyword also picks up the tweets where the keyword was used as a hashtag. A Twitter API request consists of query text (which contains keywords and operators) and parameters. To access the endpoint, a bearer token associated with the developer app you created with your account has to be sent in the request header. The query text is as follows:

```
"(father OR dad OR husband OR boyfriend) prochoice lang:en"
"(father OR dad OR husband OR boyfriend) prolife lang:en"
```

The logical operator "OR" has to be stated explicitly, while a blank space between keywords is interpreted as the logical operator "AND". To specify the language, an operator "lang:" is used within the query text, while start_time and end_time are request parameters. We will refer to the two collected datasets as prolife and prochoice. In the request we specified query text, as well as the start and end time in ISO 8601 date format (`2019-01-01T00:00:00.000Z` and `2020-01-01T00:00:00.000Z`, respectively). The search was thus limited to tweets that contain at least one of the keywords we stated as well as "prochoice" or "prolife", were written in English language and posted between the 1.1.2019 and 31.12.2019. Other request parameters are the fields we wish to receive (in Twitter API 1.0 the entire tweet object was returned; in 2.0 only id and text are returned by the default and other fields have to be specified), max number of results we want to receive in a single response and pagination token. Since the max number of tweets is capped to one hundred per request, if our query yields more data a pagination token is returned and the request should be repeated, with pagination token included in the request parameters.

The prolife dataset contains 17505 tweets, while the prochoice dataset contains 8906 tweets. We believed this to be a small number of tweets over such a large time period, especially because the topic of abortion is often trending and a topic of debate. We decided to check against the total number of tweets that contain either "prolife" or "prochoice" using the counts endpoint. The request is the same as the one used for data collection, but sent to a different endpoint.

In the same period and considering only tweets in English language, 994557 prolife and 203490 prochoice tweets were posted. Tweets from our dataset hold only 1.76% of overall prolife and 4.38% of overall prochoice tweets.

Since in further analysis we decided to use only tweets which are not of type "retweet", it is worthwhile to note that after including the "-is:retweet" operator[1] in the query there are 303193 prolife tweets, and 75034 prochoice ones. In our filtered dataset there are 5970 prolife and 3600 prochoice tweets which accounts for 1.96% and 4.9% in the overall prolife and prochoice tweets, respectively.

It seems that the topic is more prominent in the prochoice dataset. However, further analysis showed that most prochoice Tweets only contain the word "men" which is often used to denote "men" as gender, not necessarily as male partners. We decided to look at how many tweets there are if we query using the rest of the keywords without "men". There are 7799 such tweets in prolife (0.78%) and only 1222 tweets in prochoice (0.60%). If we disregard retweets, there are 2477 prolife tweets (0.82%) and 594 prochoice tweets (0.79%).

We thus concluded the following:

- prolife community is more prolific on topic of abortion on Twitter
- in both communities the discussion on the role of men is marginal

[1] "is:retweet" operator is used for fetching only tweets that are labeled as retweets. "-" is a negation sign which means "-is:retweet" fetches all tweets which are not retweets.

Keywords	Retweets (Y/N)	“men” included (Y/N)	Tweet count	Percentage in reference dataset
prolife	Y	N	994557	100%
prochoice	Y	N	203490	100%
prolife, men, father, dad, husband, boyfriend	Y	Y	17505	1.76%
prochoice,men, father, dad, husband, boyfriend	Y	Y	8906	4.38%
prolife, father, dad, husband, boyfriend	Y	N	7799	0.78%
prochoice, father, dad, husband, boyfriend	Y	N	1222	0.60%
prolife	N	N	303193	100%
prochoice	N	N	75034	100%
prolife, father, dad, husband, boyfriend	N	Y	5970	1.96%
prochoice, father, dad, husband, boyfriend	N	Y	3600	4.9%
prolife, father, dad, husband, boyfriend	N	N	2477	0.82%
prochoice, father, dad, husband, boyfriend	N	N	594	0.79%

Table 3.1 : overview of tweet count across different datasets

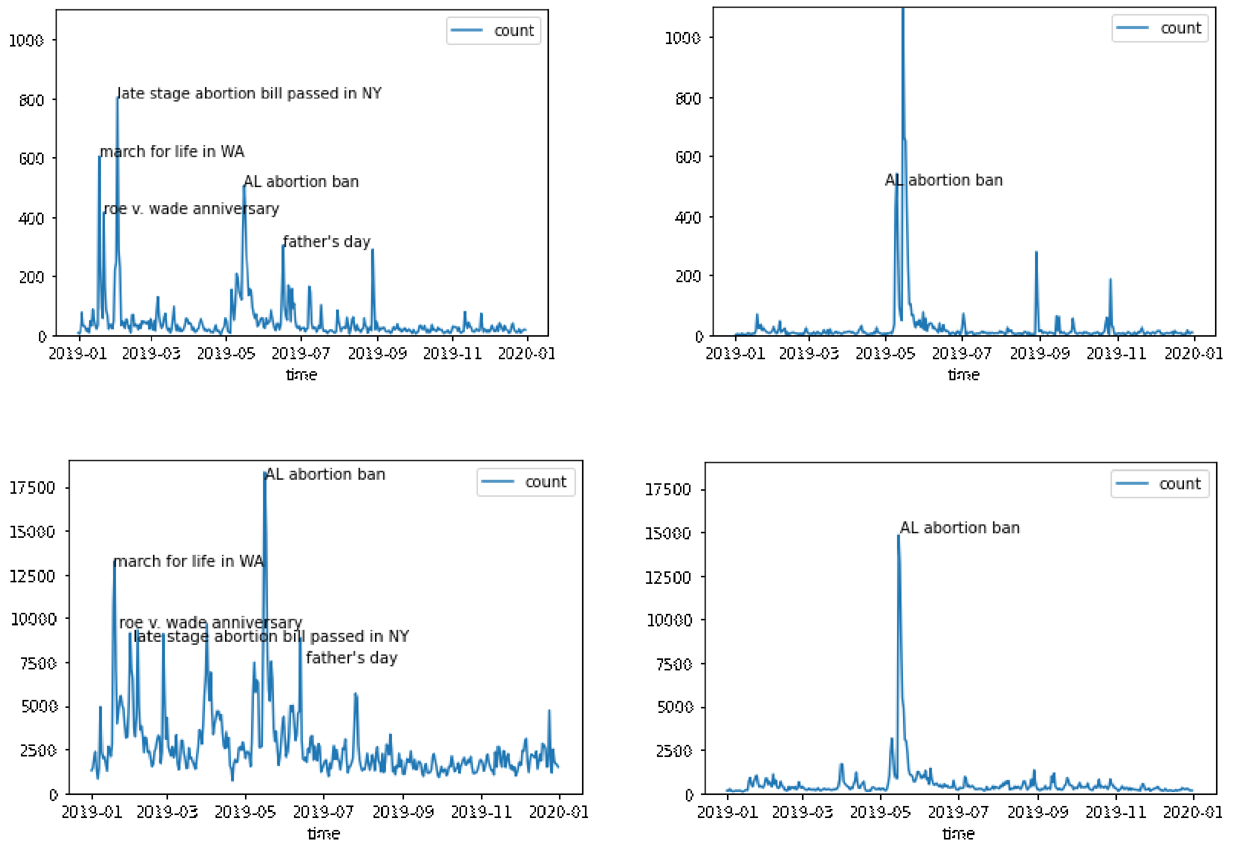


Image 3.1 : distribution of tweets over time; a) prolife (our dataset), b) prochoice (our dataset), c) prolife (overall), d) prochoice (overall)

If we look at the distribution of tweets over time we see that prolife features larger tweet volume and bigger number of peaks over time, the most notable being connected to the March for Life in Washington in January, late stage abortion bill passed in the state of New York in February, Alabama Abortion Ban in May and Father's day in June. In the prochoice data, we see one big peak in the number of tweets in May which correspond to Alabama Abortion Ban, followed by two smaller peaks in September and November which upon inspection are due to high number of retweets. This has more to do with tweet content, which is such that it garners a lot of retweets (asking for retweets or promoting something) but are not, to our best knowledge, linked to any specific event in the abortion discussion. The prochoice community seems to be more *reactive* with tweeting, as they increased tweet volume in response to abortion bans, while the prolife community was *active* throughout the year. Since the abortion is legal in most of the states, this makes sense. We can also infer that trend in our dataset follows the trends in the overall abortion discussion topic, with no larger discussion on role of men outside the ongoing abortion discussion.

Finally, we also note that not all tweets in the prolife dataset are actually prolife, and the same applies for prochoice. However, by using a different approach in data collection, such as identifying prolife/prochoice hashtags or popular users in either communities we would further diminish our dataset. Before applying network analysis, we pruned the dataset of wrongly classified tweets that had a high number of retweets. The results show that, even though there is some noise, the prolife/prochoice tweets overall reflect their respective sentiment.

We've concluded that the role of the male partner does not seem to be a prominent topic in the overall abortion data. How important are men in our dataset, which was created by specifically targeting male-related tweets? After removing hashtags, punctuation, emojis, hyperlinks, stopwords (except for possessive pronouns) and stemming the words we identified 5884 unique words in prolife and 4181 in prochoice. We counted the word frequency in tweets. Here, we displayed words that appeared in either the prochoice or prolife dataset's top twenty words, where a rank value of 0 denotes that the word is not in the dataset, while 20 is the highest rank.

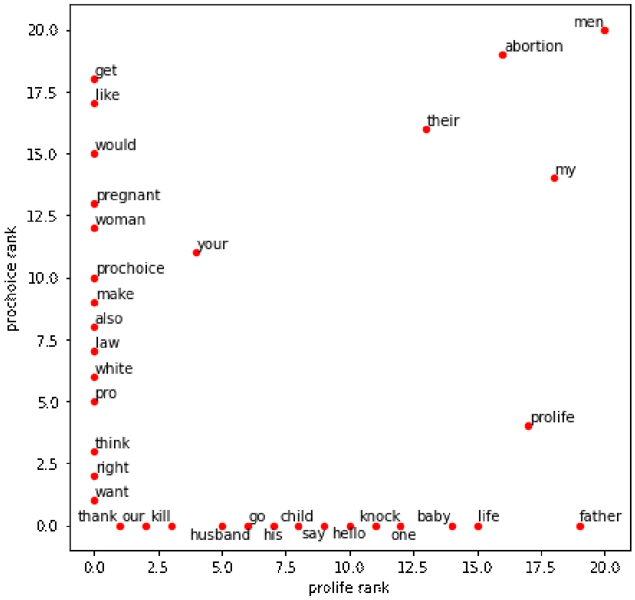


Image 3.2 : word ranking in prolife and prochoice (top 20 words ranked by frequency)

We see the words which achieved high frequency in prolife, but not in prochoice along the x-axis and those that achieved high frequency in prochoice, but not in prolife along the y-axis. Words that achieved high frequency in both are "your", "their", "my", "prolife", "abortion" and "men".

"Men", "father" and "husband" appear in the top twenty most frequent words in prolife. "Dad" was 26th and "boyfriend" 104th most frequent word. In the prochoice only "men" achieves high frequency, taking the first place. "Father" is 41st, "dad" 66th, "husband" 125th and "boyfriend" 285th most frequent word.

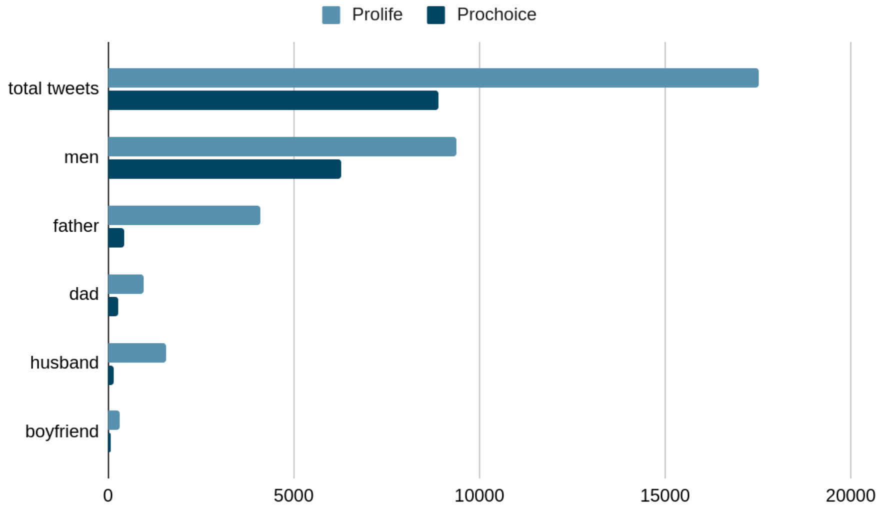


Image 3.3 : frequency of male-related keywords in the prolife and prochoice datasets

While men appear frequently in both datasets, other keywords are much more prominent in the prolife dataset. We assumed that “father” is being used in a religious connotation within the prolife dataset which was confirmed by later analysis. However, dad, boyfriend and especially husband were all percentually more represented in prolife, than prochoice.

What about hashtags?

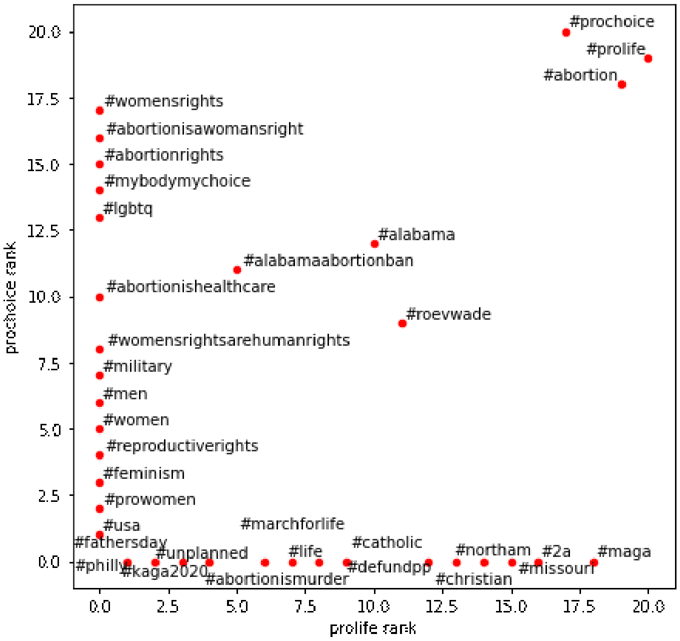


Image 3.4 : hashtag ranking in prolife and prochoice (top 20 words ranked by frequency)

Looking at hashtags might help us identify important topics in the two groups. In top twenty they have six hashtags in common: “#prolife”, “#prochoice”, “#abortion”, “#roevwade”, “#alabama” and “#alabamaabortionban”. Prochoice features hashtags which are supportive of women and their right to choose what to do with their bodies while prolife are supporting Trump, second amendment (right to bear arms), marching for life and defunding planned parenthood, claiming abortion is murder and talking about christianity. It is ironic to note that the prolife community claims abortion is murder and marches for life while supporting gun ownership. Men appear as “#men” in prochoice and as “#fathersday” in prolife.

There were 2286 unique hashtags in prolife and 1731 in prochoice. Since male-related hashtags were featured in very small number of tweets (#husband for example only once in prolife and zero times in prochoice), our decision to search by keywords instead of hashtags is justified.

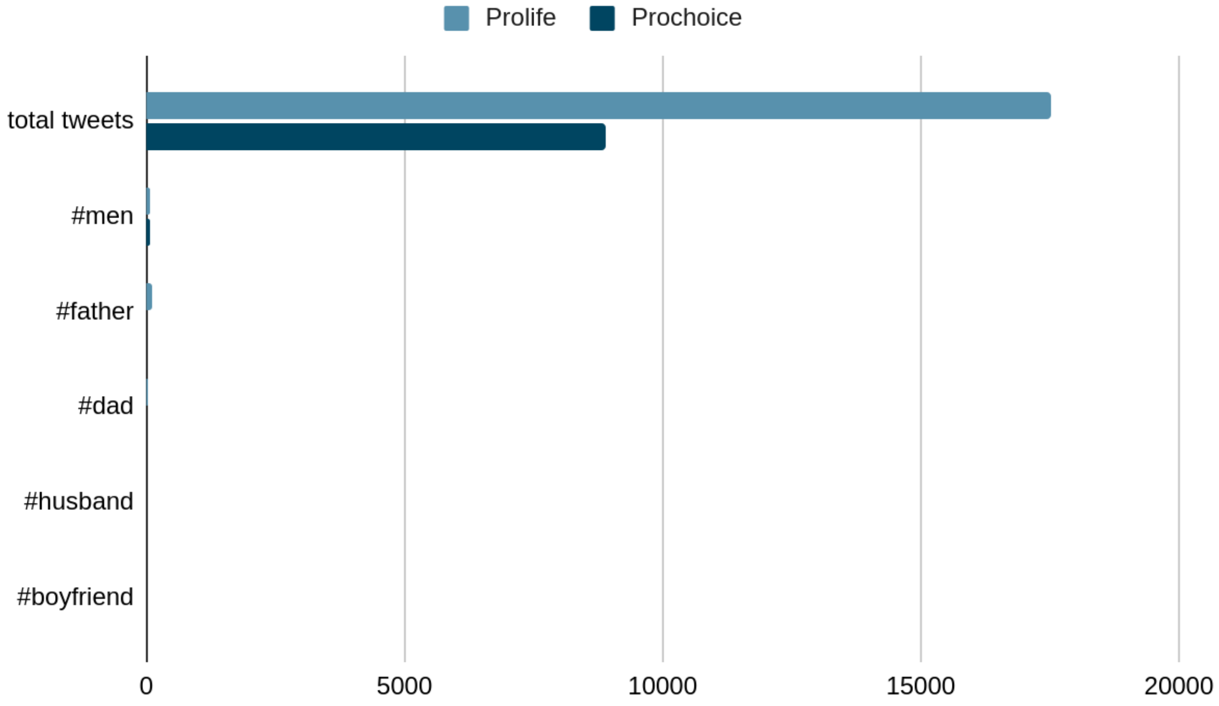


Image 3.5 : frequency of male-related keywords (as hashtags) in the prolife and prochoice hashtag datasets

A tweet can be one of four types: tweet (original status posted for the first time), retweet, reply and quote (retweet with additional text; only this added text is considered to be the tweet’s text). Out of these four types, retweet is the only one that does not introduce original content. In both datasets, retweets are the most frequent tweet types, followed by tweets.

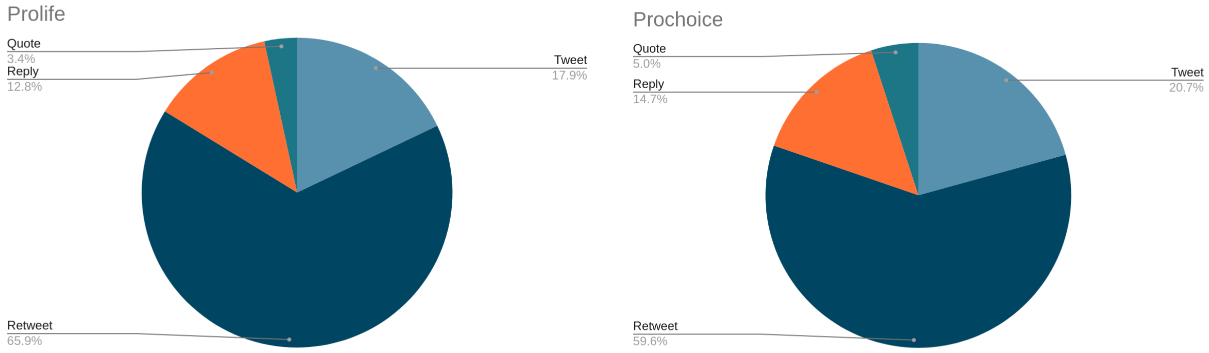


Image 3.6: percentage of tweet types in a) prolife b) prochoice dataset

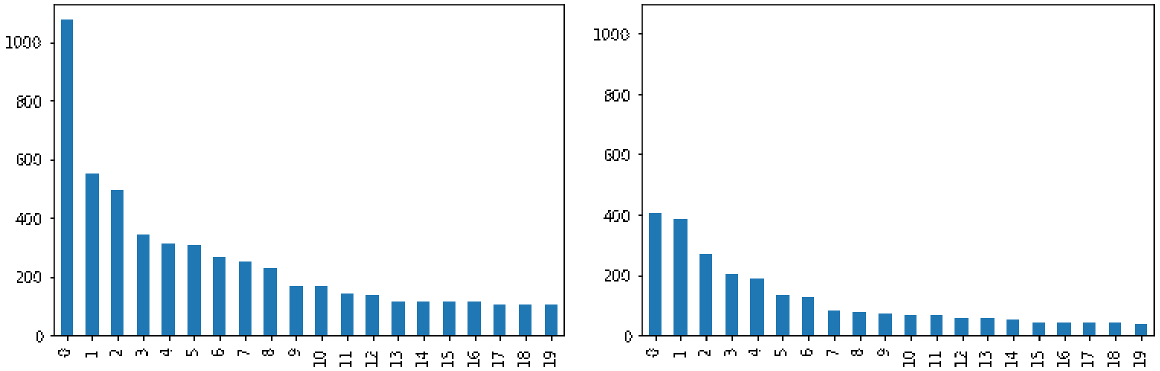


Image 3.7: Number of retweets for most retweeted tweets in a) prolife b) prochoice dataset

3.4 Cleaning the data

We kept only the original content tweets (tweet, reply and quote), weighing them by the number of retweets to account for the tweet's popularity. After initial analysis we noticed some noise in both dataset so we manually inspected tweets with more than ten retweets in order to identify wrongly labeled tweets (a tweet that contains the word "prochoice" and thus falls into the prochoice dataset but the context of the tweet is actually prolife and vice versa) and tweets irrelevant to the discussion.

In the prolife dataset, only 166 tweets out of 5970 original content tweets were retweeted more than ten times. There were 17 tweets that were clearly prochoice, and 12 irrelevant tweets. Prochoice dataset had 3600 original content tweets with 66 tweets that were retweeted more than ten times, out of which 6 were prolife and 4 were irrelevant. In prolife we identified two users whose individual tweets were not retweeted a lot but they were overall very present in the dataset, allowing their words to gain prominence. Since their posts had nothing to do with the topic[2] (they just used “prolife” to gain visibility), they were removed from the dataset. In the end we have 5306 prolife tweets, and 3589 prochoice tweets.

We cleaned the tweet text by removing hashtags, hyperlink, emojis, punctuation and stop words (except for possessive pronouns). The words were lemmatized using WordNetLemmatizer. Words that do not belong to the English language corpus were discarded. We also removed capitalisation for both words and hashtags.

3.5 Network analysis

3.5.1 Methods

A network is a graph represented by its nodes and edges (connections between the nodes). There are different analyses that can be done on the network; we decided to focus on community detection and PageRank of the nodes. We were aiming to capture the topics in the network and see how well represented, if at all, is the discussion on the role of the male partner.

One way of analyzing Twitter data through the lens of network science is by creating a semantic network using the textual contents of a tweet. The role of nodes is assigned to individual words and hashtags while edges are based on their occurrences in the dataset. Hashtags are considered especially important because they’re used as identifiers on Twitter. When a hashtag becomes popular, people tweeting on the topic or belonging to a certain Twitter subgroup use the hashtag to take part in the discussion. If we look at the most featured hashtags in our datasets, for example, we can see slogans (“#mybodymychoice”, “#abortionishealthcare”, “#kaga”), topics (“#fathersday”, “#abortion#”, “#marchforlife”, “#2a”, “#roevswade”) and group affiliations (“#feminism”, “#lgbtq”, “#christian”). These are all different ways a hashtag marks the tweet.

In API version 2.0, Twitter added a new type of metadata to the Tweets called annotations. There are two types of annotations, entity and context. Entities are based on what’s explicitly mentioned in the text, and are comprised of people, organizations, places and products. Context annotations are inferred based on the Tweet text and result in domain and/or entity labels. There are currently more than fifty domains. These annotations can be used to capture tweet’s topic more accurately than with hashtags so we will refer to them as topics.

To create a network, we tried several different approaches, all based on the concept of a bipartite network. Bipartite network consists of two types of nodes, where the edges exist only between

[2] One of them was an author promoting her young adult book, the second an aspiring influencer and third an “alternative” news account.

nodes belonging to the separate groups. We tried bipartite networks made of combinations of the three sources we mentioned (words, hashtags and topics) as well as a bipartite network where one node type is either word or hashtag and the other is a topic. We found the last approach to offer the richest interpretation but we'll include the alternatives for completeness.

We connected nodes by creating an edge between them if they appeared in the same tweet. Weight of an edge was the sum of the number of cooccurrences weighted by the number of retweets of a tweet. We removed "#prolife" from the prolife dataset and did the same for "#prochoice" in prochoice because these nodes achieve very high ranking and affect community detection. We imported the nodes and edges into Gephi, where we calculated nodes' PageRank and modularity (using Louvain's algorithm).

3.5.2 Results

3.5.2.1 Word-hashtag bipartite network

We removed tweets that do not contain hashtags, which left us with 3487 tweets in prolife and 3002 tweets in prochoice.

Network	Prolife	Prochoice
Number of nodes	6040	4764
Number of edges	50760	36818
Number of communities	12	12
Top 20 nodes	#abortion, men, #prochoice, "whywemarch, #maga, father, abortion, #roevwade, my, #uniquefromdayone, #prowoman, life, #marchforlife, #abortionismurder, #alabama, #prochild, their, #missouri, #adoptionnotabortion, #metoo	men, #prolife, #abortion, #abortionrights, #womensrightsarehumanrights, #roevwade, abortion, #womensrights, #mybodymychoice, #yesallwomen, #abotionisawomansright, #alabama, #boycottalabama, #trustwomen, get, #srhr, white, #fascism, #abortionishealthcare, their

Table 3.2 : overview of the word-hashtag networks

We display only the highest ranked nodes, in order to show words as well. We also filtered by PageRank, leaving around 200 top-ranked nodes. The color is assigned based on the size of the community, with blue assigned to the biggest community, light green to the second biggest and orange to the third biggest community in both networks.

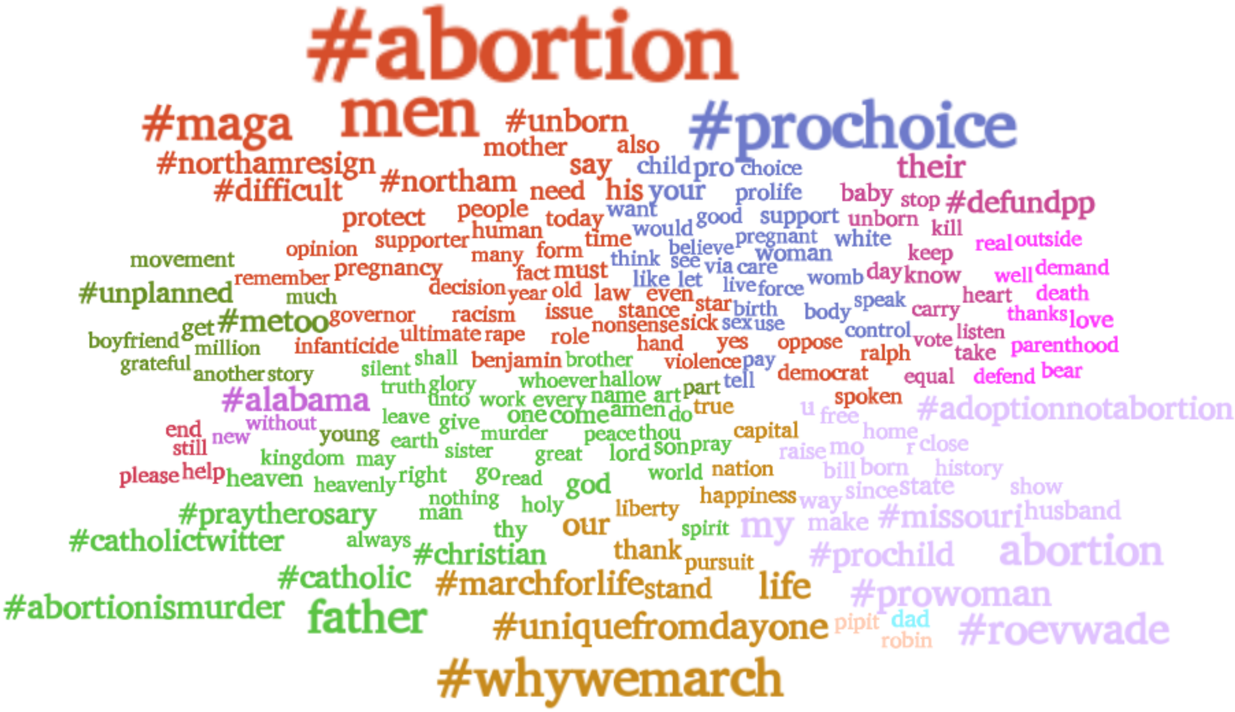


Image 3.8: prolife bipartite word-hashtag network

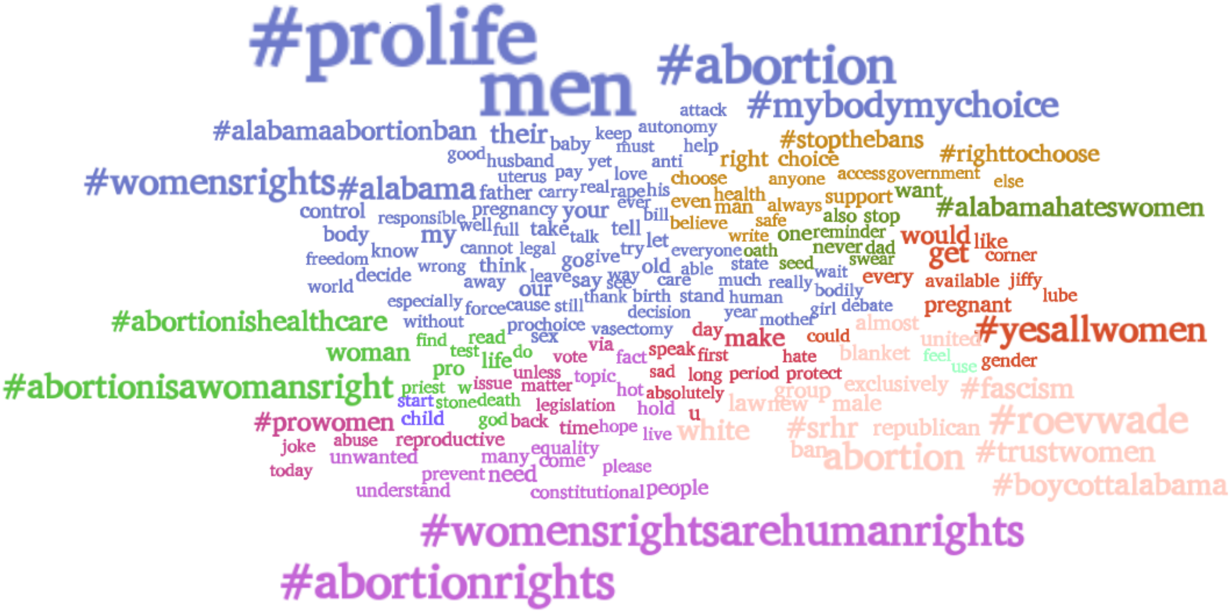


Image 3.9: prochoice bipartite word-hashtag network

If we exclude the topics from the network, the communities are not that meaningful and the role of men is not apparent in the networks. The latter finding is not supported by inspecting the tweet texts, because in the most popular tweets the discussion on what role men have in the decision making is evident. Moreover, hashtags tend to gain more prominence in these networks. We can still grasp the difference in language, especially hashtags, used in the prolife and prochoice network. Both networks have one large community (size is based by the percentage of nodes belonging to the community), the blue community (55.5% in prolife, 72.48% in prochoice). In prolife, other than the biggest community we can see a religious community (second biggest, light green), "#abortion" community (orange), March For Life community (occher), defund Planned Parenthood community[3], anti-abortion(lilac). These communities follow the tweet volume trend reported in 3.3, with biggest peaks appearing around the March For Life and different bills being passed in different US states. In prochoice, other than the biggest communities (blue and orange) we see a myriad of communities that fight for women's reproductive rights and to stop the abortion bans. They're very similar in topics which supports the trends noted in 3.3, the prochoice community becoming very active in response to Alabama abortion ban and being quite inactive for the rest of the timeline observed. In prolife "#prochoice" belongs to a different community than "men" and "#abortion" while that is not the case for "#prolife" in "#prochoice". The keywords are placed as follows. In prolife, father belongs to the religious community, where it is used to denote god. Boyfriend, dad and husband were assigned to marginal communities and it's hard to grasp the context in which these words are used. On the other hand, in prochoice, "father", "husband" and "boyfriend" (not pictured) were in the biggest community together with "men" which is not surprising as this community holds almost the entire network. Since the community detection did not yield meaningful results we did not apply further analysis of the importance of the individual communities within the network.

On the other hand, topics offer a guaranteed way to examine language used in relation to a certain topic. The downside to this approach is the fact that not all tweets are annotated, thus relying on this approach left us with 2132 prolife tweets and 1195 prochoice tweets.

We displayed only the topics which we found were the most informative, since some domains are too broad and do not add value (for example: Person, Brand, Sports, Entertainment, Family and Life Stages) while more specific ones we kept (exact name of the person, organization, athlete, actor, fatherhood and marriage). We chose to filter out irrelevant communities. Finally, nodes were also filtered by PageRank, leaving around 200 nodes per network in order to have a more readable visualization.

[3] *Planned Parenthood* Federation of America is a nonprofit organization that provides sexual health care in the United States and globally.

3.5.2.2 Word-topic bipartite network

Network	Prolife	Prochoice
Number of nodes	4510	3094
Number of edges	54155	33608
Number of communities	8	6
Top four communities (topic and percentage of nodes inside the community)	politics (43.95%), fatherhood (26.3%), Planned Parenthood (13.7%), marriage (11.95%)	politics (48.8%), Planned Parenthood (19.81%), fatherhood (19%), miscellaneous (12.09%)
Top 20 nodes (after filtering out the irrelevant topics)	Fatherhood, men, Donald Trump, life, father, Planned Parenthood, my, Ted Cruz, prolife, thank, Alyssa Milano, protect, their, Ralph Northam, your	men, #prolife, #abortion, #abortionrights, #womensrightsarehumanrights, #roevwade, abortion, #womensrights, #mybodymychoice, #yesallwomen, #abotionisawomansright, #alabama, #boycottalabama, #trustwomen, get, #srhr, white, #fascism, #abortionishealthcare, their

Table 3.3 : overview of the word-topic networks

Since this approach yielded meaningful topics, we assigned the community color based on the topic (blue: politics, green: fatherhood, purple: Planned Parenthood and in the case of prolife orange: marriage). We filtered out irrelevant communities (communities 5-8 in prolife, 4-6 in prochoice).

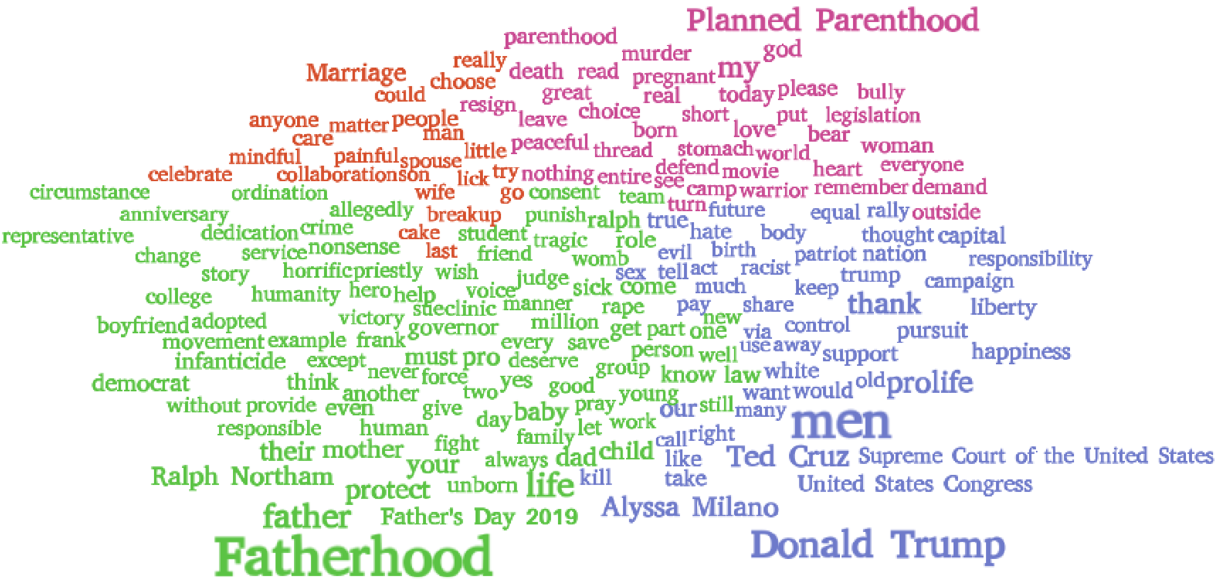


Image 3.10: prolife bipartite word-topic network

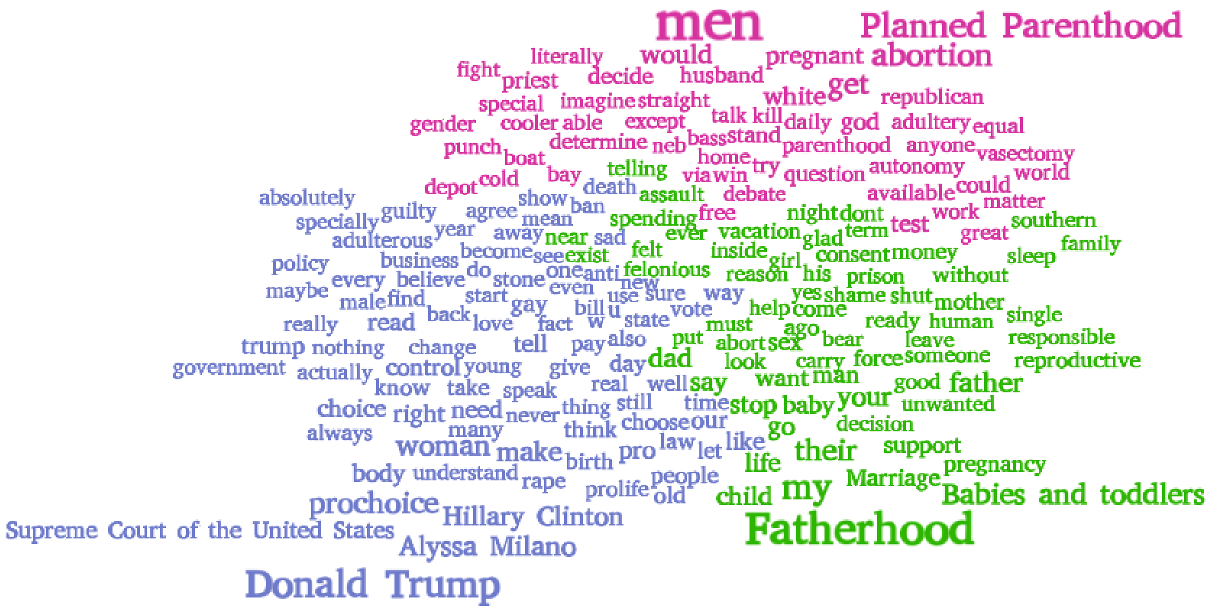


Image 3.11: prochoice bipartite word-topic network

Most prominent communities in prolife and prochoice are talking about the same topic (Politics, Fatherhood, Planned Parenthood and Marriage), albeit in different ways.

In both datasets, the community which holds the biggest percentage of network (in terms of number of nodes belonging to the community) is the blue community, which can be dubbed the "politics" community since it features nodes such as Donald Trump, Hilary Clinton, Supreme Court of the United States and United States Congress. This comes as no surprise, since in a similar study we've previously done the US politics dominated the abortion discourse. Donald Trump has himself engaged in dangerous rhetoric regarding abortion which has fired up both his supporters and his critics. However, in prolife we can see "men" and "prolife", pro-American words (mentioning liberty and pursuit of happiness, nation, capital, patriot) while in prochoice's blue community the words are focused on "woman", "birth", "right" and choice "prochoice", "choice", "choose", Both datasets feature Alyssa Milano in the blue community since she invited women to go on a "sex strike" in response to strict abortion laws and prolife/prochoice.

The green, "Planned Parenthood" community is the second largest in prochoice and third largest in prolife. Planned Parenthood is another strongly polarizing topic. While prolife activists try to defund the organization and often protest in front of the facilities, verbally abusing people who seek their services, the prochoicers strongly support Planned Parenthood as a place that offers affordable reproductive healthcare. This is reflected in the nodes assigned to the communities. In the prolife network it's filled with negative words such as murder, death, bully. Prochoice associates words such as "men", "pregnant", "get", "abortion", "autonomy", "need", and "vasectomy".

The topic of Fatherhood, which is of biggest interest for this study, holds the purple community. It is the second largest community in prolife, and third in prochoice. It is interesting to note that "Marriage" topic was in the same community in prochoice, while in prochoice it is a separate community. Comparing the node size, we can see that "Fatherhood" achieved a higher ranking in prolife community. Other highly ranked nodes inside the prolife Fatherhood community are "father", "life, "dad", "protect", "baby", "child", "mother", "Father's day 2019" and "Ralph Northam"[4]. The sentiment on the topic of abortion can be grasped from words such as "infaticide", "unborn". In prochoice, the community is smaller and the nodes belonging to it achieve lower centrality than in the prolife network. Highly ranked is the topic of "Babies and toddlers", possessive pronouns "my", "your", "their", words such as "father", "dad", "baby", "child", "support", "reproductive", "man", and "sex".

The Marriage topic is in a separate community inside the prolife network, taking fourth place by size. It is associated with words "choose", "mindful", "collaboration" and "man".

[4] Ralph Northam is a democrat Virginia governor who sparked outrage for allegedly supporting late-term abortions. He later faced a scandal when his yearbook photo showing a man in a Ku Klux Klan costume and another in blackface was leaked online.

In prolife, “prolife” and “men” were assigned to the politics community. “Father”, “dad” and “boyfriend” were assigned to the fatherhood community, while “husband” was in a marginal community. “Prochoice” was also in the politics community, but with lower rank (not displayed). In the prochoice network, “men” and “husband” were associated with the Planned Parenthood community. On the other hand, “father” and “dad” were associated with the fatherhood community. “Prochoice” and “prolife” were in the politics community. “Boyfriend” was in the miscellaneous community (not pictured).

3.5.2.3 Hashtag-topic bipartite network

There were only 1437 prolife and 947 prochoice tweets that had both hashtags and topics.

Network	Prolife	Prochoice
Number of nodes	1778	1139
Number of edges	9492	5981
Number of communities	7	11
Top four communities (topic and percentage of nodes inside the community)	politics (33.35%), fatherhood (29.75%), miscellaneous (14.12%), Planned Parenthood (12.6%)	politics (34.77%),Fatherhood (22.56%), Planned Parenthood (14.93%), miscellaneous (12.64%)
Top 20 nodes (after filtering out the irrelevant topics)	#maga, Fatherhood, Babies and toddlers, #whywemarch, #unborn, #uniquefromdayone, Donald Trump, #metoo, #prochoice, Ted Cruz, #alabama, Planned Parenthood, #fathersday, #abortionismurder, #chooselife	#prolife, Planned Parenthood, #abortion, #mybodymychoice, #plannedparenthood, #abortionisawomansright, Fatherhood, #womensrights, #roevwade, Babies and toddlers, Donald Trump, #alabamahateswomen, #privacy, #abortionishealthcare, #alabamaabortionbill, #womensrightsarehumanrights

Table 3.4 : overview of the hashtag-topic networks

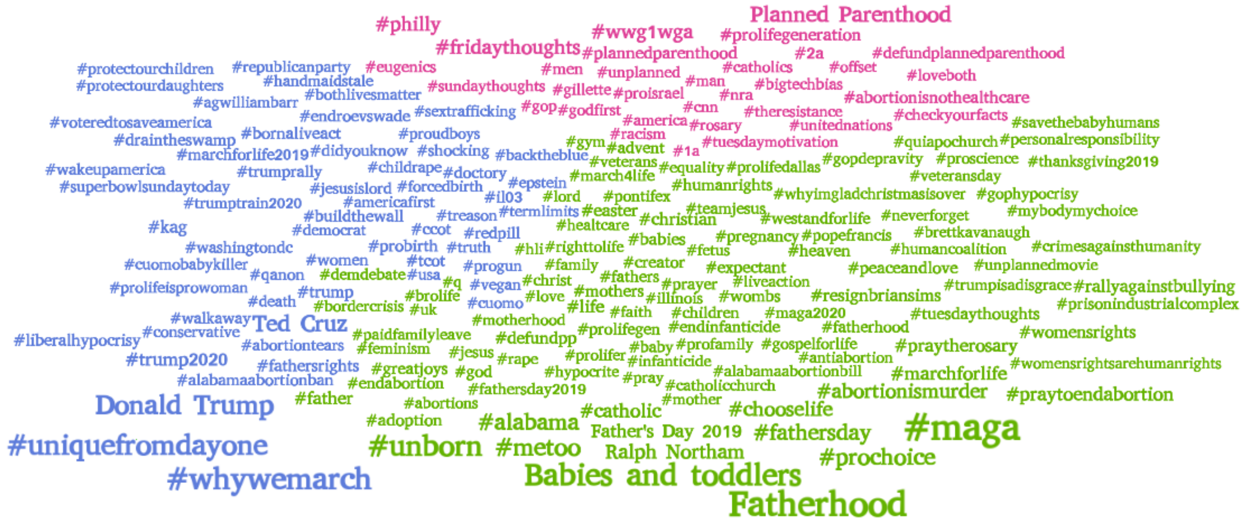


Image 3.12: prolife bipartite hashtag-topic network

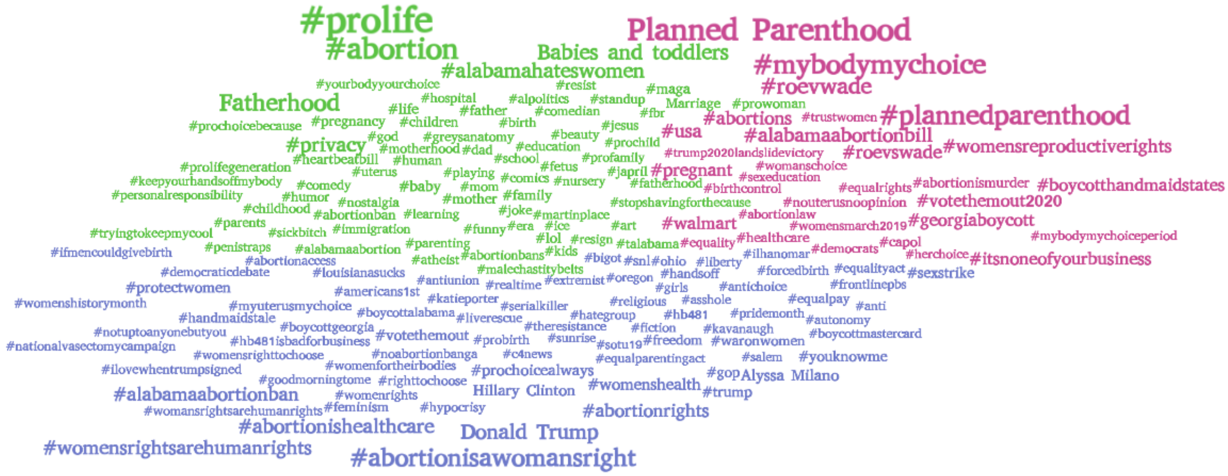


Image 3.13: prochoice bipartite hashtag-topic network

Again, the "politics" community (featuring Donald Trump in both networks, Ted Cruz in prolife and Hillary Clinton in prochoice) is the biggest community in both networks. Fatherhood is second biggest in both, while Planned Parenthood is the fourth and third in prolife and in prochoice, respectively. The topic of Marriage (not pictured) was assigned to the politics community in prolife and Fatherhood community in prochoice. Both networks had a miscellaneous community in the top 4, which was irrelevant to the discussion and thus filtered out.

The hashtags are reflective of the group interests, similar to the results in 3.4.2.1. There are no prominent hashtags that somehow include men in the abortion discussion, except for "#fathersright" in prolife and "#ifmencouldgivebirth" in prochoice, both in the politics community.

3.5.2.4 Word+hashtag-topic bipartite network

Network	Prolife	Prochoice
Number of nodes	5819	3851
Number of edges	63744	39589
Number of communities	7	8
Top four communities (topic and percentage of nodes inside the community)	politics (41.09%), fatherhood (29.3%), miscellaneous (12.37%), Planned Parenthood (12.1%)	politics (48%), Planned Parenthood (19.24%), Fatherhood (18.8%), marriage (6.39%)
Top 20 nodes	Fatherhood, men, Donald Trump, abortion, life, father, Planned Parenthood, #abortion, stand, his, my, say, prolife, thank, Alyssa Milano, Ralph Northam, protect, star, time	men, Fatherhood, Donald Trump, Planned Parenthood, my, abortion, babies, get, prochoice, their, woman, alyssa, father, make, hillary, dad, your, would, say, child

Table 3.5 : overview of the word+hashtag-topic networks

In both datasets, the community which holds the biggest percentage of network (in terms of number of nodes belonging to the community) is the blue community, which can be dubbed the "politics" community since it features nodes such as Donald Trump, Hilary Clinton, Supreme Court of the United States and United States Congress.

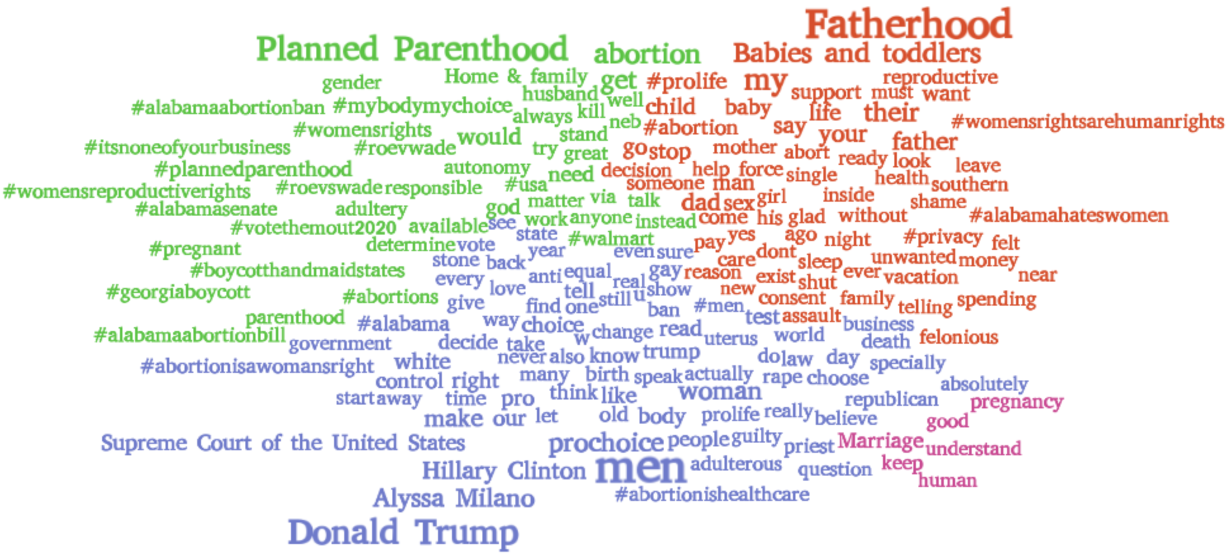


Image 3.14: prolife bipartite word+hashtag-topic network

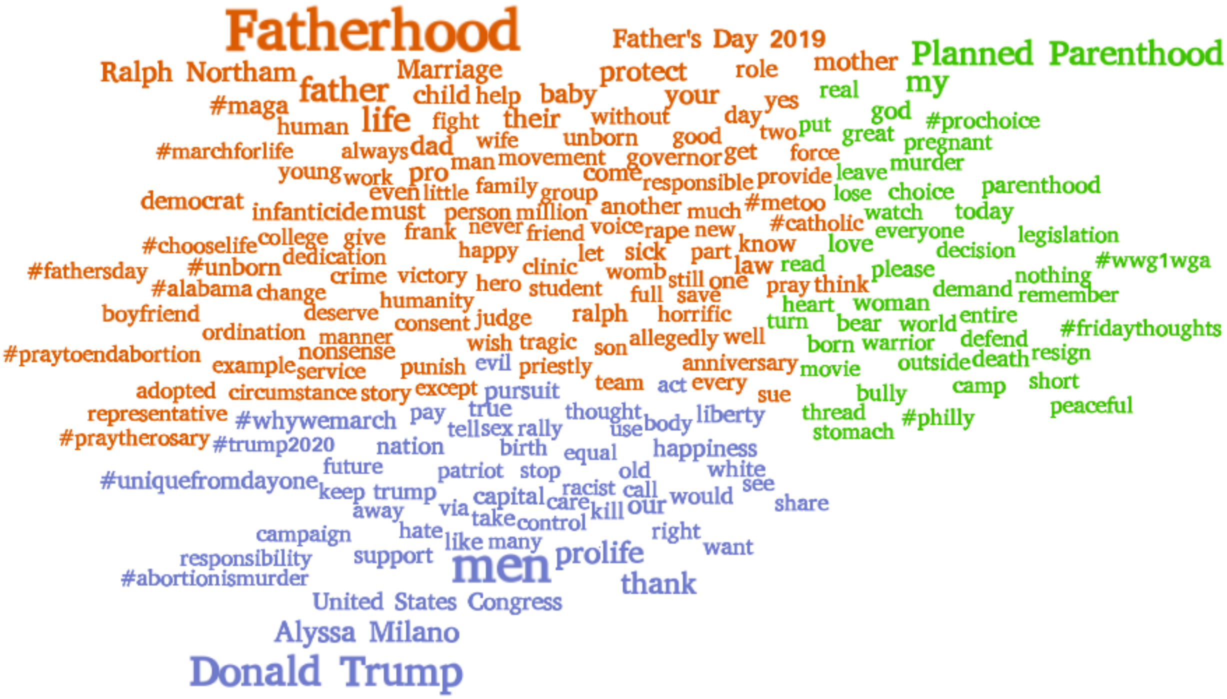


Image 3.15: prochoice bipartite word+hashtag-topic network

However, in prolife we can see anti-abortion sentiment with hashtags “#abortionismurder”, “#whywemarch” and “#uniquefromdayone” (a March for Life slogan) and pro-American words (mentioning liberty and pursuit of happiness, nation, capital, patriot) while in prochoice’s blue community there are hashtags “#abortionisawomansright”, “#abortionishealthcare” and “#alabama”, no patriotic sentiment, Hilary Clinton and focus on fighting the Alabama ban (“#alabama”, “Supreme Court of the United States”, “ban”). Both datasets feature Alyssa Milano in the blue community since she invited women to go on a “sex strike” in response to strict abortion laws and prolife/prochoice.

The green, “Planned Parenthood” community is the second largest in prochoice and fourth largest in prolife. The conflicting sentiment of the two groups toward the Planned Parenthood is reflected in the nodes assigned to the communities. In the prolife network it’s filled with negative words such as murder, death, bully and “#wwg1wga” a hashtag used by the Qanon[5] members. Prochoice associates the topic of “Home and family” with Planned parenthood, as well as hashtags “#womensreproductiveright”, “#itsnoneofyourbusiness”, “#mybodymychoice” and words such as “autonomy”, “need”, and “abortion”.

The topic of Fatherhood, which is of biggest interest for this study, holds the orange community. It is the second largest community in prolife, and third in prochoice. It is interesting to note that “Marriage” topic was in the same community in prolife, while in prochoice it is a separate community. Comparing the node size, we can see that “Fatherhood” achieved a higher ranking in prolife community. Other highly ranked nodes inside the prolife Fatherhood community are “father”, “life, “dad”, “protect”, “role”, “baby”, “child”, “mother”, “Father’s day 2019” and “Ralph Northam”. The sentiment on the topic of abortion can be grasped from words such as “infaticide”, “unborn”, “crime” and hashtags “#praytoendabortion”, “#chooselife”, “#marchforlife”. Religion-related hashtags (“#praytherosary”, “#catholic”, pray) are present here as well, reflecting the duality of the word “father”, which is used to denote the male parent and god. This is apparent in the prolife supporters, who overall tend to be conservative, religious and right-leaning. In prochoice, the community is smaller and the nodes belonging to it achieve lower centrality than in the prolife network. Highly ranked is the topic of “Babies and toddlers”, possessive pronouns “my”, “your”, “their”, words such as “father”, “dad”, “baby”, “child”, “support”, “reproductive”, “stop”, and “sex”. Hashtags associated with this community are “#womensrightsarehumanrights”, “#privacy”, “#abortion”, and “#alabamahateswomen”. Even though the highly ranked nodes are similar in the both communities we can see by closer inspection of nodes that even though there is talk on role of fathers in the prochoice network, it is still talked about in the context of female reproductive rights whereas in prochoice it is put in the context of fathers fighting the abortion, family values (such as marriage and celebrating father’s day) and, marginally, in the context of religion. We can also contrast words “protect” in prolife with “support” in prochoice. In the prolife network the word “protect” is close to the topic of fatherhood, and “support” is inside the political community. In prochoice “support” is close to the fatherhood, while “protect” is inside the Planned Parenthood community (not displayed due to low PageRank score). Talking about the Fatherhood community, we can also notice subtle

[5] *QAnon* is an American far-right political conspiracy theory and mass political movement.

differences by contrasting “unborn” with “unwanted” in prochoice, “provide” in prolife and “pay” in prochoice.

The Marriage topic is in a separate community inside the prochoice network, taking fourth place by size. It is associated with words “good”, “pregnancy”, “keep” and “human”. The smaller communities were filtered out in both networks. In prolife we also filtered out the third largest community as it was miscellaneous and had no relation to the topic of abortion.

Now we take a look at positioning of our keywords within the network. “Father” and “dad” got assigned to the fatherhood community in both networks. “Husband” was a part of a marginal community in prolife and Planned Parenthood community in prochoice. “Boyfriend” is also in the fatherhood community within prolife, and in a marginal community in prochoice. “Prolife” and “prochoice” belong to the political community in both networks (although “prolife” is more highly ranked in prolife network, and so is “prochoice” in prochoice network). Interestingly, the hashtag “#prochoice” is associated with Planned Parenthood in prolife, while the hashtag “#prolife” is associated with the Fatherhood community in prochoice. This can be interpreted as both groups talking about the other party in relation to what they deem is important to them (and this sentiment is true, because Fatherhood is the second most important community in prolife, while Planned Parenthood is the second most important community in prochoice). Finally, “men” are assigned to the political community in both networks, underlining the fact the word is often used to denote gender (men as politicians, men as law makers) and not just men as male partners, although this usage appears in the dataset as well.

3.6 Conclusion

The discussion on the role of men is marginal in the overall discussion on the topic of abortion on Twitter. That is apparent from the low percentage of tweets on abortion containing male-related words, as shown in the first section. The analysis was done on the original dataset, which was further diminished later on by cleaning spam accounts and irrelevant tweets that achieved high number of retweets. Another indicator is the absence of any prominent hashtags related to men in either of the datasets that would signal an ongoing movement to include men in the discussion. Conversely, in the prolife dataset the community rallies around anti-abortion hashtags, while in prochoice the hashtags have to do with women's (reproductive) rights.

We tried four different approaches to building the networks. In the word-hashtag network, male partners as a topic were undetectable, which contradicted manual inspection of the most retweeted (and thus most prominent) tweets in the dataset. However, it validated the quality of data in two ways: by showing a clear difference between the prolife and prochoice network and by having most prominent communities reflect the topics assigned to peaks in the tweet volume.

However, by relying on tweet topics, which were a part of metadata provided by Twitter, we managed to show that the topic of fatherhood is present and relevant in both datasets. Topic-based networks identified three important topics in both dataset: politics, fatherhood and Planned Parenthood, with their prominence varying depending on the dataset and the way the network was built. Another relatively important topic was Marriage, which was sometimes a community of its own, and other times was assigned to the other three communities. Out of the three networks (word-topic, hashtag-topic and word+hashtag-topic), the hashtag-topic network had the weakest results as it again failed to capture the context in which male-related words are used and it had only two men-related prominent hashtags (#fathersrights in prolife, #ifmencouldgivebirth in prochoice). The results of the other two are similar to each other, but the mixed network (words+hashtags-topics) offered the richest interpretation, where the sentiment around a topic was carried by words and underlined with prominent hashtags.

By observing this network, we have shown that the topic of fatherhood is more prominent in the prolife dataset, both by the number of nodes assigned to it (29.3% in prolife versus 18.8% in prochoice) and by the PageRank of the topic itself (0.018 in prolife, which makes it the highest rated node in the network versus 0.014 in prochoice which makes it the second highest ranked in the network).

By analyzing words close to the topic of fatherhood we have concluded that in prochoice men are seen as those who should fight against the abortion and protect their unborn babies while in prochoice they are seen as those who should support the woman and her right to chose how to handle an unwanted pregnancy. We can conclude that both groups include male partners in the discussion only if their opinions are in line with the group sentiment and no concern is given to male partner's individual thoughts, feelings and agency outside of it.

4. Analyzing political discourse before and after election using Tweets posted by the candidates for the 2020 US Elections

4.1 Project introduction and goals

Before elections politicians try to persuade their potential voters. We aimed to quantify this phenomenon and analyze how politicians change the content and style of their communication on social media. We looked at eight different cases based on whether the tweet was posted before or after elections, by a chosen member or politician who lost the election and finally whether it was posted by a democrat or a republican. Using network science tools, we identified topics relevant to each of the scenarios and measured how prominent these topics are in the dataset as a whole. The aim was to see how the discourse changes over time and across party lines. We also look at how politicians change the content and style of their communication on social media depending on the actual phase of the election cycle, measuring agency and concreteness of the shared messages.

4.2 Software tools

For data collection, cleaning and network building we used Python 3.7.12.

To scrape the names of the candidates, we used BeautifulSoup, a Python library for web scraping.

To lookup profiles based on the first and last name of the members, we used the Twitter API version 1.1 endpoint `/users/search.json`. A standard access Twitter developer profile enables sending unlimited requests to this endpoint, with the only limitation being that of the number of requests (nine hundred requests per fifteen-minute time window).

To collect the users' tweets we communicated with the official Twitter API, version 2.0, using the Python requests library. We used an account with Academic research access, which allows us to fetch up to 10 million tweets per month, access to all endpoints, query parameters and operators as well as the entire historical archive of Twitter. We collected users' timeline using `/users/{id}/tweets`.

To clean the text we used nltk (natural language toolkit) and re (regex) libraries. For data storing and manipulation we relied on the pandas library.

To create network graphs from our data we used networkx, a network science library in Python.

For statistical visualization, we relied on matplotlib. To visualize network graphs, calculate PageRank and perform community detection we used Gephi, an open source software.

4.3 Data collection and preliminary analysis

Elections in the United States are held every other year, on even years. They are held on the first Tuesday in November. Last election date at the time of data collection was November 3, 2020. Congress has two constitutive parts:

- House of Representatives, which is constituted of:
    - 435 voting members
    - 6 non-voting members, representatives of US territories such as Virgin Islands, which we will disregard
- Senate, which is constituted of 100 members.

Generally, every election all seats of the House of Representatives are up for voting as well as one-third of the senators, because they are elected for 6-year terms that overlap in such a way that every second year some of the seats are to be re-elected. Republicans and Democrats put forward one or more people for each slot. They choose the candidate among their members but that is beyond the scope of this discussion. Other parties nominate their members, and there can also be independent candidates. We will focus only on the republican and democratic candidates, as the others didn't win any seats in 2020 elections.

In 2020:

- 35 senators were elected, 33 because of the regular proceedings and 2 were elected to fill vacancies (one senator died, the other resigned). Democrats won 15 spots, Republicans 20.
- 435 representatives were elected, 222 democrats and 213 republicans

It is worthwhile to note that these numbers are constantly changing throughout the mandate, with elected members resigning their post for various reason but that is irrelevant to the discussion. We base our study on the immediate result of the elections.

The data collection process was different from the one described in the third chapter. We were interested in all tweets posted by specific users, and not tweets posted by any user but based on a specific topic. The first step was to find the names of the candidates, both the ones that were elected and their opponents who lost. The second step was to find their Twitter usernames. Final step was to collect their timelines between two specified dates (nine month prior to and nine month post the election date).

4.3.1 Elected members

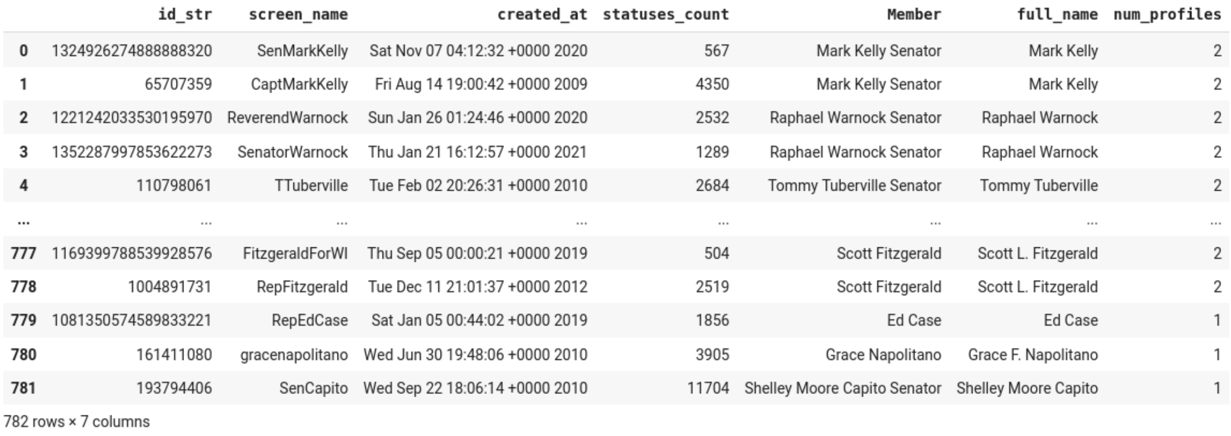
We have used BeautifulSoup library to scrape the names of members who were elected to the HoR and Senate from Wikipedia. For every member, we have noted their full name, the party they belong to and the role they assumed (Senator or Representative).

We used the user search API endpoint to look for Twitter profiles of elected members using their full names and roles they assumed as the search query. For the House of Representative members we used the full name together with keywords “representative”, “congress”, “congressman, “congresswoman” while for the members of the Senate we used “senator”.

We then pruned the data, removing profiles which had less than 10 tweets, and then inspecting it manually and removing junk profiles further. After collecting tweets, we deleted information about profiles who have not posted in the observed period.
Five members did not have an active Twitter profile. Another five have not had more than 10 posts in their feed. We ended up with 461 members who have one or more valid Twitter profiles (out of 470 members). Total profile count was 782.

Out of those numbers there were 7 members with three Twitter profiles, 307 members who had two Twitter profiles, and 147 members with one Twitter profile.

For every profile we recorded the id, screen name, date of profile creation, number of tweets posted, full name, full name with role and number of profiles.



	id_str	screen_name	created_at	statuses_count	Member	full_name	num_profiles
0	1324926274888888320	SenMarkKelly	Sat Nov 07 04:12:32 +0000 2020	567	Mark Kelly Senator	Mark Kelly	2
1	65707359	CaptMarkKelly	Fri Aug 14 19:00:42 +0000 2009	4350	Mark Kelly Senator	Mark Kelly	2
2	1221242033530195970	ReverendWarnock	Sun Jan 26 01:24:46 +0000 2020	2532	Raphael Warnock Senator	Raphael Warnock	2
3	1352287997853622273	SenatorWarnock	Thu Jan 21 16:12:57 +0000 2021	1289	Raphael Warnock Senator	Raphael Warnock	2
4	110798061	TTuberville	Tue Feb 02 20:26:31 +0000 2010	2684	Tommy Tuberville Senator	Tommy Tuberville	2
...	...	...	...	...	...	...	...
777	1169399788539928576	FitzgeraldForWI	Thu Sep 05 00:00:21 +0000 2019	504	Scott Fitzgerald	Scott L. Fitzgerald	2
778	1004891731	RepFitzgerald	Tue Dec 11 21:01:37 +0000 2012	2519	Scott Fitzgerald	Scott L. Fitzgerald	2
779	1081350574589833221	RepEdCase	Sat Jan 05 00:44:02 +0000 2019	1856	Ed Case	Ed Case	1
780	161411080	gracenapolitano	Wed Jun 30 19:48:06 +0000 2010	3905	Grace Napolitano	Grace F. Napolitano	1
781	193794406	SenCapito	Wed Sep 22 18:06:14 +0000 2010	11704	Shelley Moore Capito Senator	Shelley Moore Capito	1

782 rows × 7 columns

Image 4.1: twitter profile data of the elected members

4.3.2 Candidates who lost the election

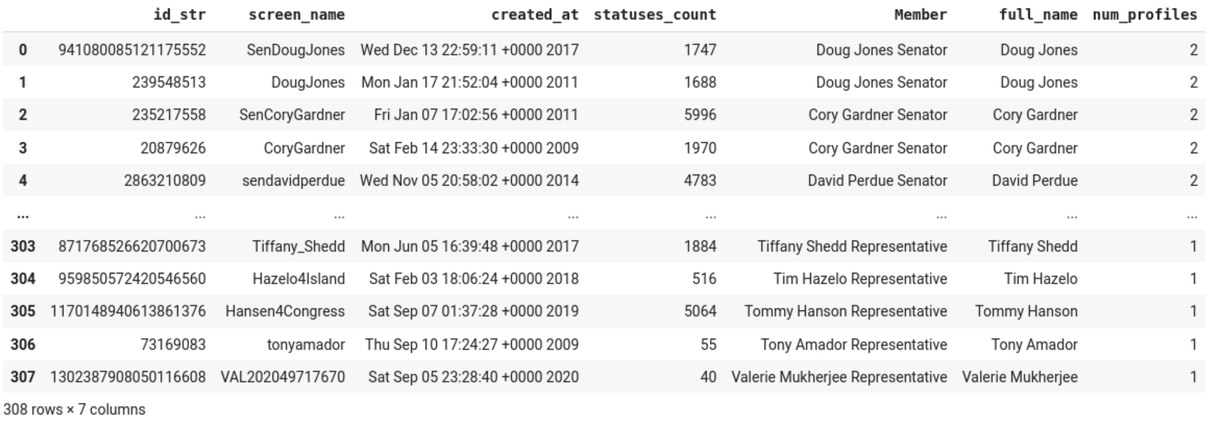
We used the user search API endpoint to look for Twitter profiles of members who lost their election in the same way as described in the previous section. Even though the candidates who lost the election are less likely to be associated with keywords we used previously (such as “congressmen” and “senator”), we decided not to perform the search based on full name only, because when we validated that approach we ended up with too many irrelevant profiles. Unfortunately, many members were not picked up with this search so we had to find them manually.

After collecting all the existing profiles, to our best knowledge and cross-checking with Ballotpedia[4], pruning the fake profiles and profiles who had fewer than 10 tweets, deleting information about profiles who have not posted in the observed period, we ended up with 372 candidates who had one or two valid Twitter profiles (out of 465 candidates). Total profile count was 388 profiles.

Out of those numbers there were 16 candidates with two Twitter profiles and 356 candidates with one Twitter profile.

It is worth noting that some candidates who ran for the election and lost used to have a Twitter account that was in the meantime deleted. Some of them were banned by Twitter and for the others we can assume that they deleted the profile after the elections because they used it for campaign purposes only.

For every profile we recorded id, screen name, date of profile creation, number of tweets posted, full name, full name with role and number of profiles.



	id_str	screen_name	created_at	statuses_count	Member	full_name	num_profiles
0	941080085121175552	SenDougJones	Wed Dec 13 22:59:11 +0000 2017	1747	Doug Jones Senator	Doug Jones	2
1	239548513	DougJones	Mon Jan 17 21:52:04 +0000 2011	1688	Doug Jones Senator	Doug Jones	2
2	235217558	SenCoryGardner	Fri Jan 07 17:02:56 +0000 2011	5996	Cory Gardner Senator	Cory Gardner	2
3	20879626	CoryGardner	Sat Feb 14 23:33:30 +0000 2009	1970	Cory Gardner Senator	Cory Gardner	2
4	2863210809	sendavidperdue	Wed Nov 05 20:58:02 +0000 2014	4783	David Perdue Senator	David Perdue	2
...	...	...	...	...	...	...	...
303	871768526620700673	Tiffany_Shedd	Mon Jun 05 16:39:48 +0000 2017	1884	Tiffany Shedd Representative	Tiffany Shedd	1
304	959850572420546560	Hazelo4Island	Sat Feb 03 18:06:24 +0000 2018	516	Tim Hazelo Representative	Tim Hazelo	1
305	1170148940613861376	Hansen4Congress	Sat Sep 07 01:37:28 +0000 2019	5064	Tommy Hanson Representative	Tommy Hanson	1
306	73169083	tonyamador	Thu Sep 10 17:24:27 +0000 2009	55	Tony Amador Representative	Tony Amador	1
307	1302387908050116608	VAL202049717670	Sat Sep 05 23:28:40 +0000 2020	40	Valerie Mukherjee Representative	Valerie Mukherjee	1

308 rows × 7 columns

Image 4.2: twitter profile data of the candidates who lost the election

4.3.3 Collecting Tweets

We used the user timeline search API v2 endpoint using ids of the profiles we have collected. We also specified date boundaries, between 09.02.2020. and 09.08.2021 (nine months before and after the elections).

For elected members we collected 1228391 tweets and for candidates who lost the elections we collected 409022 tweets. We can see that the elected members dataset has more than three times the number of tweets of the other dataset. The reason for it lies in fewer number of profiles (782 "winners" versus 388 "losers") and more posting by the chosen members.

4.4 Data cleaning

We cleaned the tweet text by removing hashtags, hyperlink, emojis, punctuation and stop words (except for possessive pronouns). The words were lemmatized using WordNetLemmatizer. Words that do not belong to the English language corpus were discarded. We also removed capitalisation for both words and hashtags.

We assumed that since the Tweets were posted by the US politicians there was no need to specify the language operator in the query. However, in the dataset there were a number of tweets in other languages (most notably, Spanish) and tweets with undefined value of the language. This usually happens when there is not enough text to determine the language, because the tweet contained only hyperlinks, hashtags, and/or emojis and then was left with no content after data cleaning.

Out of 1228391 tweets posted by chosen members, 1199251 were in English. For the candidates who lost the election, out of 409022 tweets, 381572 were in English.

A tweet can be one of four types: tweet (original status posted for the first time), retweet, reply and quote (retweet with additional text; only this added text is considered to be the tweet's text). Out of these four types, retweet is the only one that does not introduce original content.

In the chosen members dataset, the most common tweet type is tweet (51.2%), followed by retweet (24.2%), quote (13.2%) and reply (11.4%) . In the candidates who lost the election, the most common tweet type is the tweet (37.3%), followed by reply (24.8%), retweet (23.0%) and quote (14.8%).

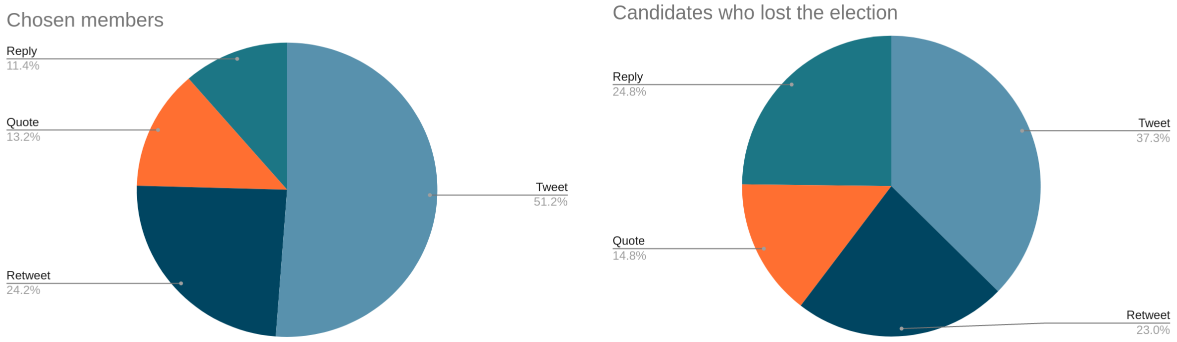


Image 4.3: percentage of tweet types in a) chosen members b) candidates who lost the election dataset

In both datasets, the prevailing type of tweet is the original status. However, in chosen members it holds over fifty percent of the dataset. It seems that the winners were more focused on producing content, than interacting with others. An interesting difference is the number of replies. In the winner dataset it is the least represented type of tweet, while in the loser dataset it is the second most represented type of tweet. These numbers indicate that the candidates who lost were more focused on interacting with others, especially by replying to and retweeting them.

After initial analysis we noticed some noise in the losers dataset so we manually inspected suspicious profiles and removed those that were picked up by the profile search but were not actually belonging to a candidate, ending up with 381572 tweets in that dataset.

We decided to keep only the original content tweets (tweet, reply and quote) in order to capture the user's unique expression. In contrast to what was done in the abortion study, we did not weigh them by the number of retweets, because it is possible (and likely) that the retweets are outside of the dataset, i.e that they were retweeted by people other than their fellow candidates and therefore the number of retweets does not reflect the tweet's popularity within our datasets. In the winner dataset, there were 908552 original content tweets, while in the losers dataset there were 293632 original content tweets.

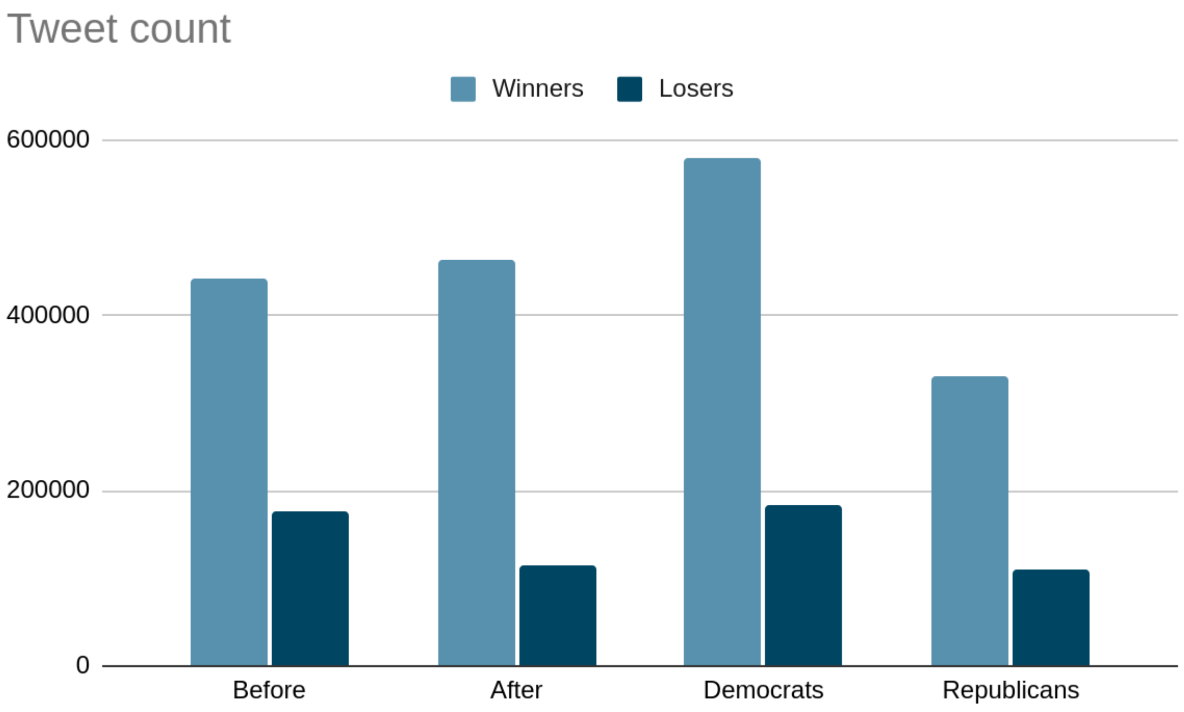


Image 4.4: tweet count across different time period, party affiliation within winner and loser datasets

When comparing tweet count trends before and after the elections, we can see a slight increase in the number of tweets posted by the winners and a decrease in the number of tweets posted by the candidates who lost the election. We excluded the tweets from the "after" datasets which were posted exactly on the election date (3532 in winners, 2296 in losers). When comparing democrats versus republicans, we can see that in both datasets democrats posted more than the republicans.

4.5 Network analysis

4.5.1 Methods

We built a bipartite semantic network where nodes of the network were either hashtags or words and edges were based on their co-occurrences. There are different analyses that can be done on the network; we decided to focus on community detection and PageRank of the nodes. We were aiming to capture the topics in the network and compare how these topics change over time, across political parties and in successful versus unsuccessful candidates.

After trying different approaches, we found that the word projection of a word-hashtag bipartite network offers the most meaningful community detection. In this network, two words were connected between each other if they both appeared with the same hashtag at least five times. The threshold was set to limit the number of edges, otherwise the analysis is computationally infeasible.

We created eight separate networks:

- chosen_members_democrat_before,
- chosen_members_democrat_after,
- chosen_members_republican_before,
- chosen_members_republican_after,
- losers_democrat_before,
- losers_democrat_after,
- losers_republican_before and
- losers_republican_after.

In each network we ran the community detection and identified the biggest six communities. We named the communities and labeled them according to the network they belong to. We then compared the 48 communities between each other to see how they correlate.

Finally, we took the 48 communities and scored their similarity to the eight relevant scenarios to see which topics are more prominent in what scenario.

After the initial run, we took into consideration the Capitol attack, which happened on 06.01.2021. We realized that the magnitude of the event would greatly affect the sentiment in the tweets that happened afterwards, especially the level of agency in the tweets. Since our analysis of the communities is done in order to study agency across different communities we decided it is best to exclude tweets that were posted after the Capitol attack.

This left us with 88209 after election tweets in winners and 21224 in losers.

4.5.2 Results

Network	Number of nodes	Number of edges	Number of communities
chosen_members_democrat_before	3583	1484967	53
chosen_members_democrat_after	1257	98336	21
chosen_members_republican_before	2592	882930	56
chosen_members_republican_after	3133	549893	14
losers_democrat_before	2003	587603	50
losers_democrat_after	3459	541284	15
losers_republican_before	2068	682706	58
losers_republican_after	3505	784960	14

Table 4.1: overview of the eight networks in terms of size (number of nodes and edges) and number of communities

One trend we can notice in the node assignment to the community is that in all four “before” networks there are more communities than in their “after” counterparts. In the next section we will show top ranked nodes in each of the networks. The nodes are sized according to their PageRank and colored according to the community they belong to. In all eight networks, the same color scheme is applied, based on the community size (starting from the biggest community: dark violet, light violet, blue, light lilac, yellow, red). Community size is the percentage of the nodes that belong to that community. We have also shown the community PageRank which is calculated as the sum of the PageRank of nodes belonging to the community. Generally, in the “before” networks, the PageRank corresponds to the size, whereas in the “after” communities there are some discrepancies (i.e a community might be biggest by number of nodes, but its PageRank would be smaller than that of the smaller communities).

4.5.2.1 Chosen members/democrats/before

Community name	Percentage (in node numbers)	PageRank
agentic and pandemic	37.51%	0.21
economical recovery	9.85%	0.01
economy	9.41%	0.09
race and history	9.18%	0.09
politics	4.47%	0.04
voting	4.19%	0.05

Table 4.2: overview of the communities in the chosen_members_democrat_before network

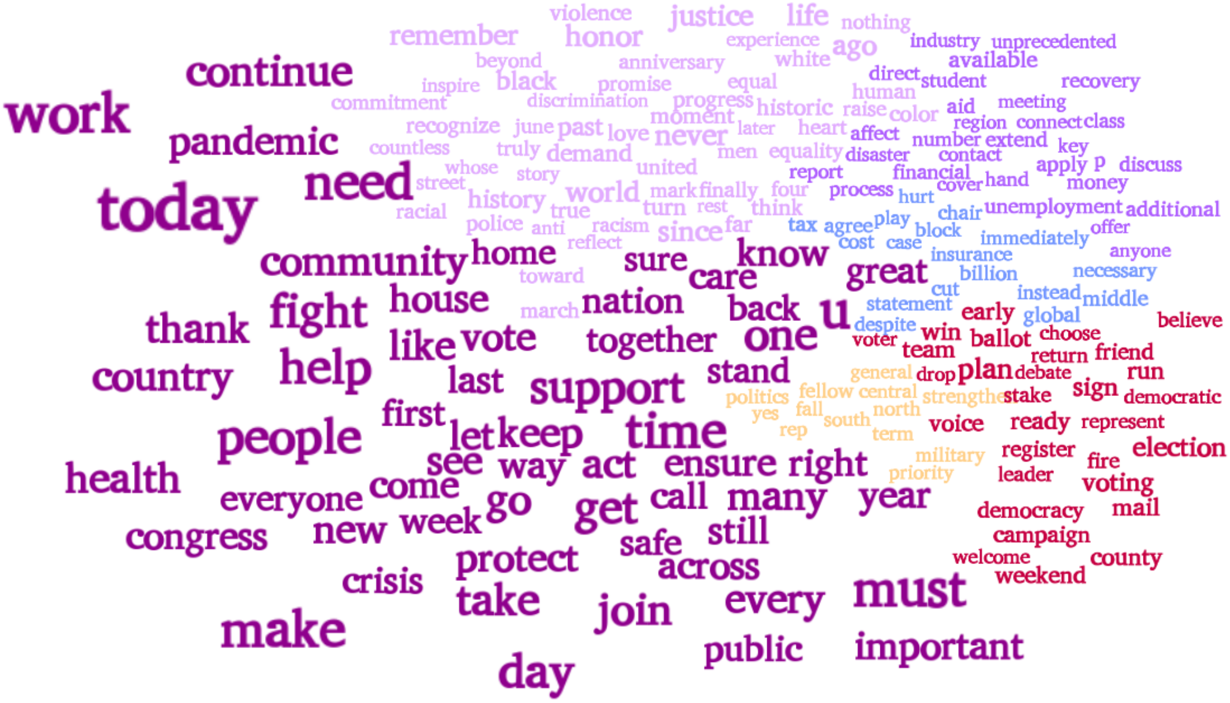


Image 4.5: communities in the chosen_members_democrat_before network

4.5.2.2 Chosen members/democrat/after

Community name	Percentage (in node numbers)	PageRank
agentic and pandemic	13.37%	0.08
agentic and race and history	12.73%	0.01
president and economy and climate	11.85%	0.12
economical recovery	8.67%	0.11
agentic and social injustice	7.96%	0.05
holidays	7.72%	0.09

Table 4.3: overview of the communities in the chosen_members_democrat_after network

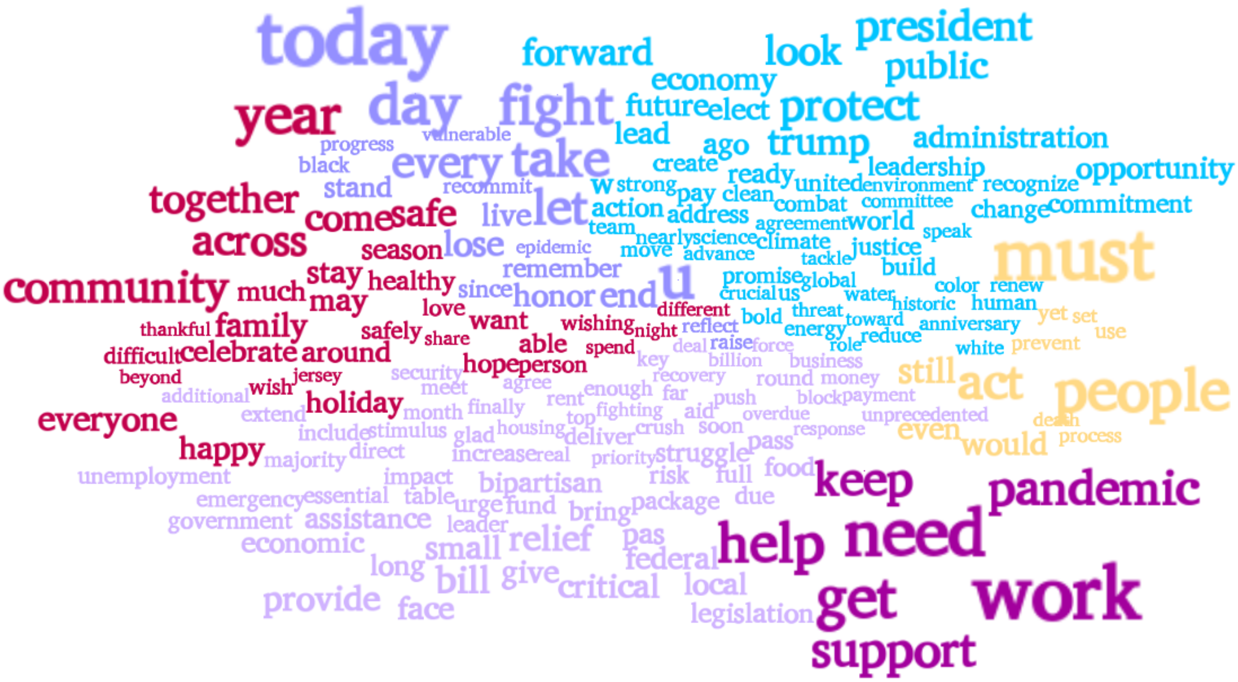


Image 4.6: communities in the chosen_members_democrat_after network

4.5.2.3 Chosen members/republican/before

Community name	Percentage (in node numbers)	PageRank
agentic and patriotic	35.22%	0.20
economical recovery	15.05%	0.15
voting	10.8%	0.12
political	7.06%	0.07
economy and development	6.37%	0.06
history and military	4.67%	0.05

Table 4.4: overview of the communities in the chosen_members_republican_before network

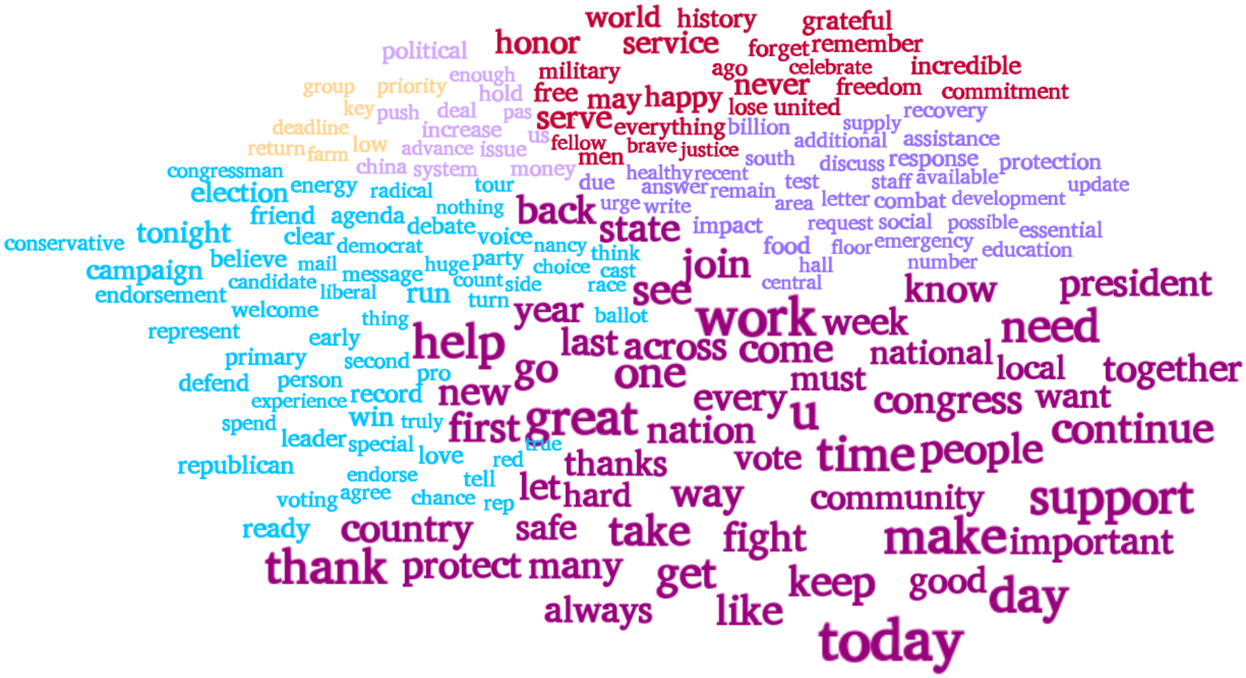


Image 4.7: communities in the chosen_members_republican_before network

4.5.2.4 Chosen members/republican/after

Community name	Percentage (in node numbers)	PageRank
pandemic and economical recovery	15.86%	0.18
history and military	15%	0.14
president	8.92%	0.11
economy and development	8.58%	0.09
unemployment	8.53%	0.09
agentic	8.28%	0.06

Table 4.5: overview of the communities in the chosen_members_republican_after network

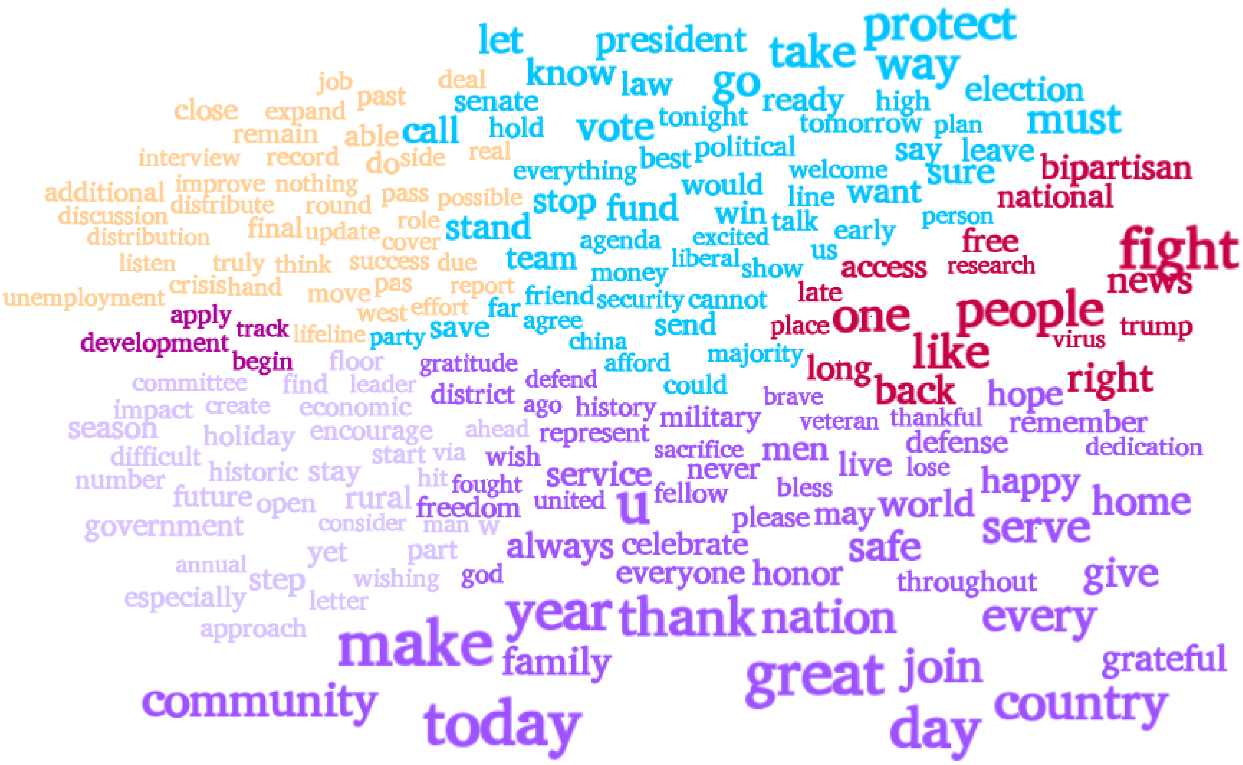


Image 4.8: communities in the chosen_members_republican_after network

4.5.2.5 Losers/democrat/before

Community name	Percentage (in node numbers)	PageRank
agentic and voting	32.35%	0.19
economical recovery and pandemic	12.28%	0.11
political debate	7.29%	0.07
voting	4.99%	0.06
economy and development	4.24%	0.04
economy and climate	4.14%	0.05

Table 4.6: overview of the communities in the losers_democrat_before network

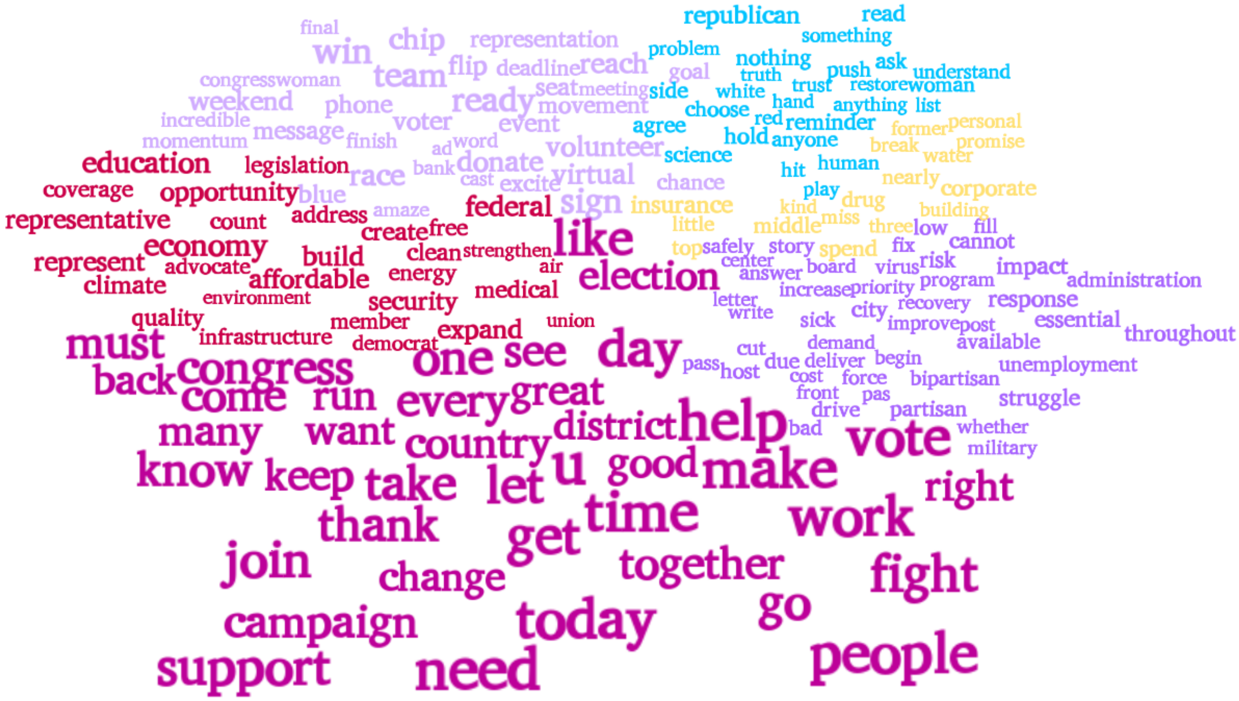


Image 4.9: communities in the losers_democrat_before network

4.5.2.6 Losers/democrat/after

Community name	Percentage (in node numbers)	PageRank
agentic and voting	21.71%	0.19
president	15%	0.13
political	14.4%	0.24
economy and community	9.74%	0.07
new year	8.41%	0.06
patriotism and development	8.3%	0.08

Table 4.7: overview of the communities in the losers_democrat_after network

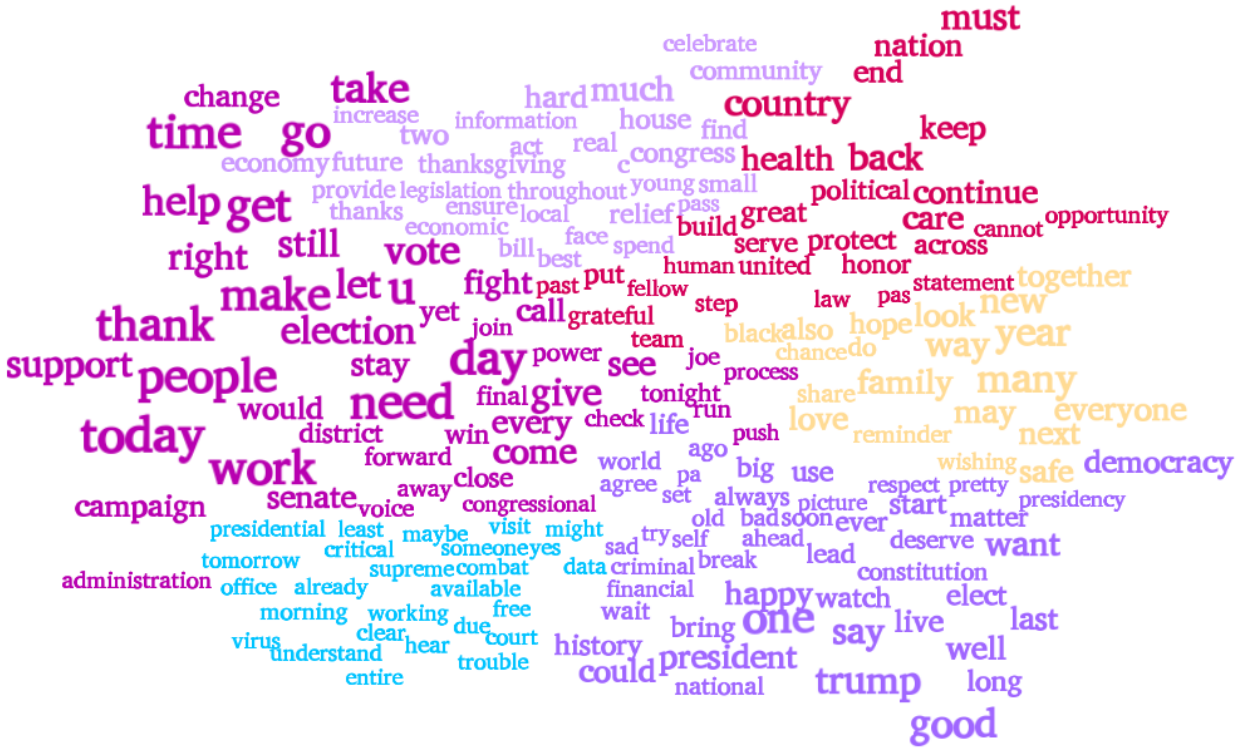


Image 4.10: communities in the losers_democrat_after network

4.5.2.7 Losers/republican/before

Community name	Percentage (in node numbers)	PageRank
agentic and voting	30.51%	0.20
racial and violence issues	13.3%	0.12
political	10.2%	0.09
activism	5.75%	0.06
political and debate	5.22%	0.06
voting	4.93%	0.04

Table 4.8: overview of the communities in the losers_republican_before network

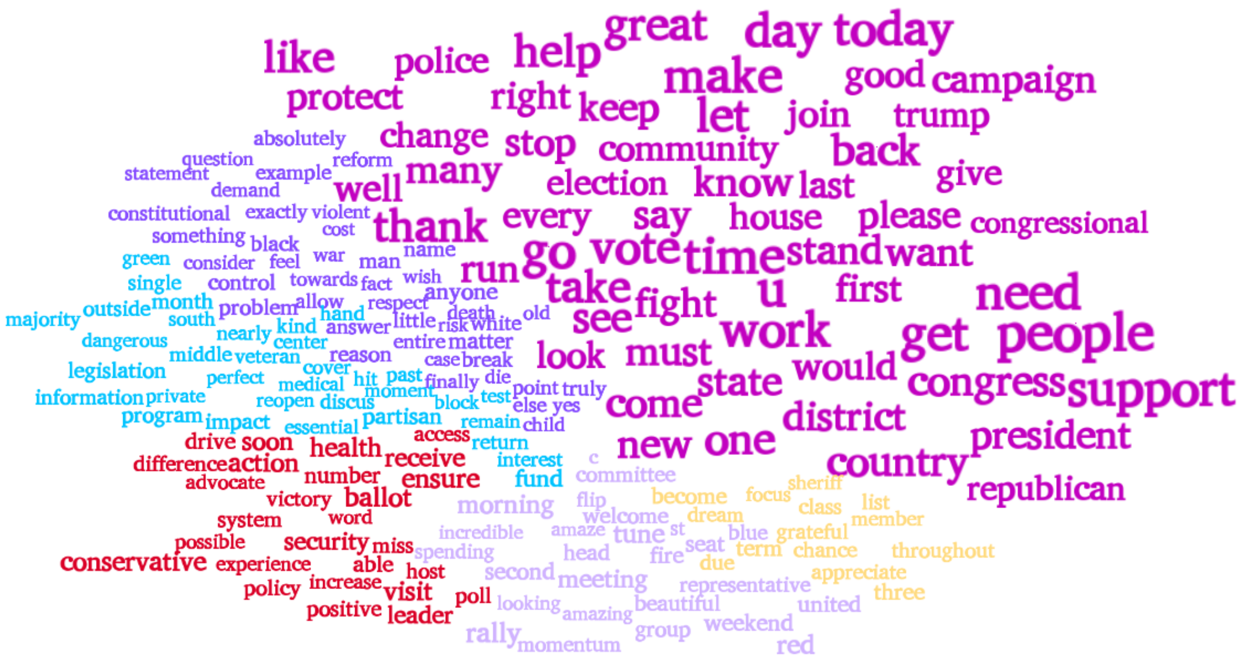


Image 4.11: communities in the losers_republican_before network

4.5.2.8 Losers/republican/after

Community name	Percentage (in node numbers)	PageRank
agentic and president	21.65%	0.18
political	13.07%	0.10
political turmoil	13.04%	0.23
economy and pandemic	11.27%	0.12
political and debate	10.64%	0.11
news and leadership	8.13%	0.08

Table 4.9: overview of the communities in the losers_republican_after network

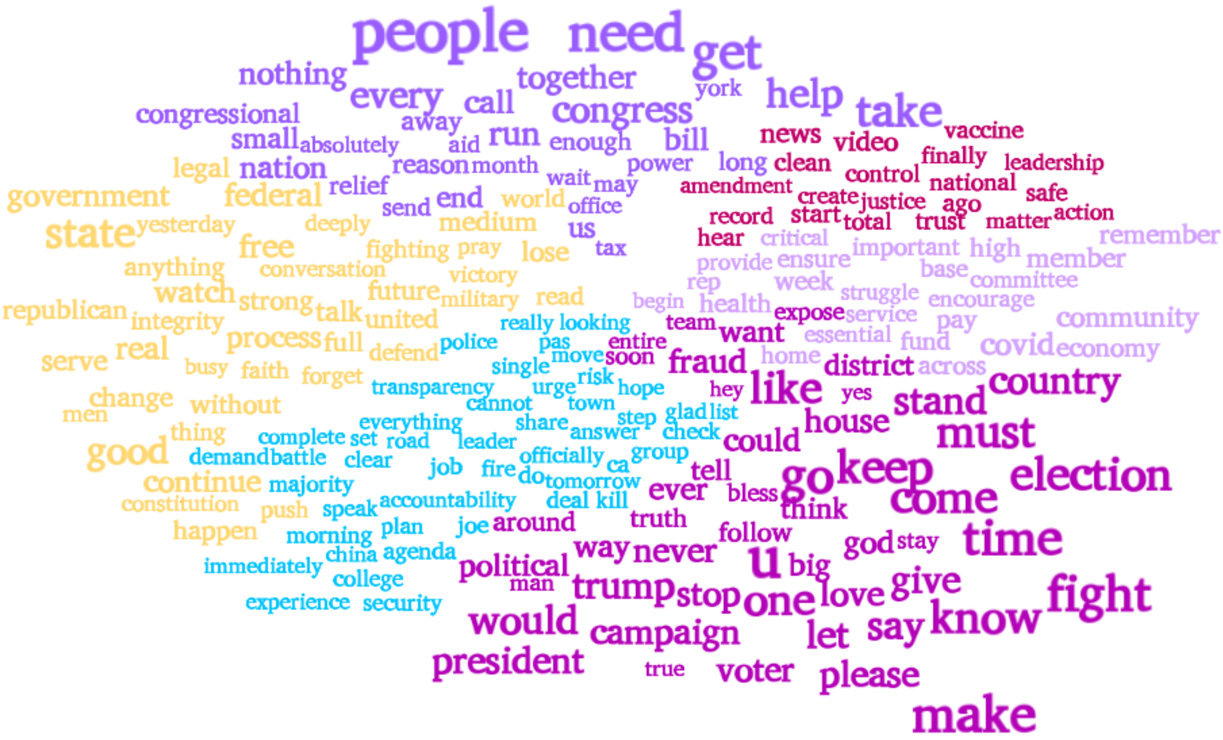


Image 4.12: communities in the losers_republican_after network

4.5.2.9 Discussion

When comparing before and after by observing community size we can conclude that the discussion gets more fragmented after the election, that is, even though there are fewer communities there is no one dominant community. While in the "before" networks, there is one larger community followed by smaller ones, the difference in size is not that prominent in the communities in the "after" network, which all tend to be similar size. This is especially true in the winner networks, where the difference between the first and second largest community is 0.64% in democrats and 0.86% in republican. However, even though the communities are smaller, they are not more focused on a specific topic; they even become more vague than the communities in the "before networks". As we have stated before, the size and importance (i.e sum of nodes' PageRank) do not correspond in the "after" networks.

We can see that some communities are common across networks, most notably the "agentic" community, dubbed thus because it is filled with agentic words such as "go", "fight", "support", "work", "make", etc. Other common topics are economy, development and economic recovery in relation to the pandemic measures. Voting-related words are a community in the before networks as well as the "after" communities, but only for the "losers". The president is mentioned in the after networks reflecting the turmoil over Trump supporters not accepting the presidential election results and claiming voter fraud. Another prominent topic is history, which in the republican networks refers to military history (honoring veterans and their service) while in the democrat networks it is in relation to the fight for racial equality. To reflect this similarity between communities, we calculated the community correlation as a sum of the PageRank of the nodes they have in common and visualized the correlation matrix. The size and the intensity of the color indicate stronger correlation.

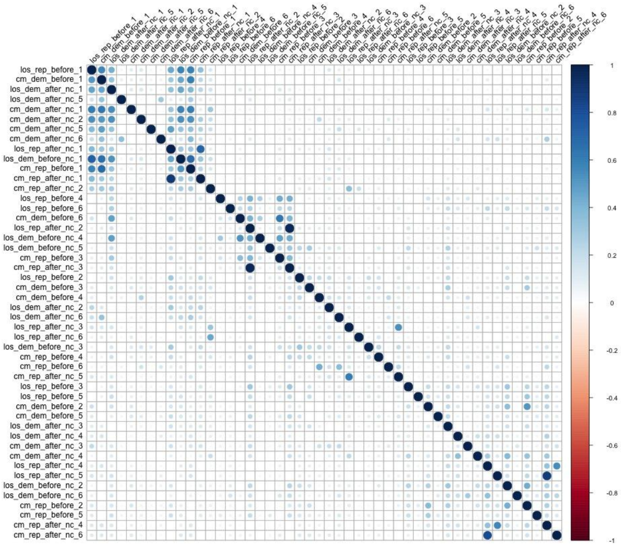


Image 4.13: community correlation matrix

To get a clearer view on the prominence of topics within different networks, we named the communities based on their topic and the source network. We calculated their PageRank score within every network by summing the value of the community nodes' PageRank within a network.
demc winns before
We then gave them a score on six parameters:

- before (sum of PageRank in the four "before" networks),
- after (sum of PageRank in the four "after" networks),
- loser (sum of PageRank in the four "loser" networks),
- winner (sum of PageRank in the four "winner" networks),
- democrat (sum of PageRank in the four "democrat" networks),
- republican (sum of PageRank in the four "republican" networks)

Then we used the average of the opposing scores (before-after, loser-winner, democrat-republican) to assign three dimensions to the data: time, outcome and party. The before, loser and republican networks were assigned these values with a negative sign. The networks were given maximum or minimum of the scores (for example, a before democrat winner network would be given a minimum time, maximum outcome and maximum party score). We created a bipartite network consisting of topic and network nodes. These nodes were assigned time, outcome and party parameters.

We used it as the input to the MDS (Multidimensional Scaling) layout algorithm which takes a matrix in which the distances between a set of entities are recorded and creates a low-dimensional (two-dimensional in our case as that is enabled by the MDS algorithm in Gephi) spatial configuration in which the same entities are located in a way that the distances between them are proportional to the distances in the original distance matrix. [5]

We used time as one dimension (x-axis) and election outcome as the second dimension (y-axis).

To accurately represent the third dimension, we used the Network Splitter 3d [6] layout, and used the party affiliation as the third ("z") dimension. Since it is hard to position the textual nodes in three dimensions accurately (due to label overlap), we used the result of the Network Splitter 3d algorithm to color the nodes. Red represents the republican party, blue represents the democratic party while the nodes who are in between these two values are assigned a violet color.

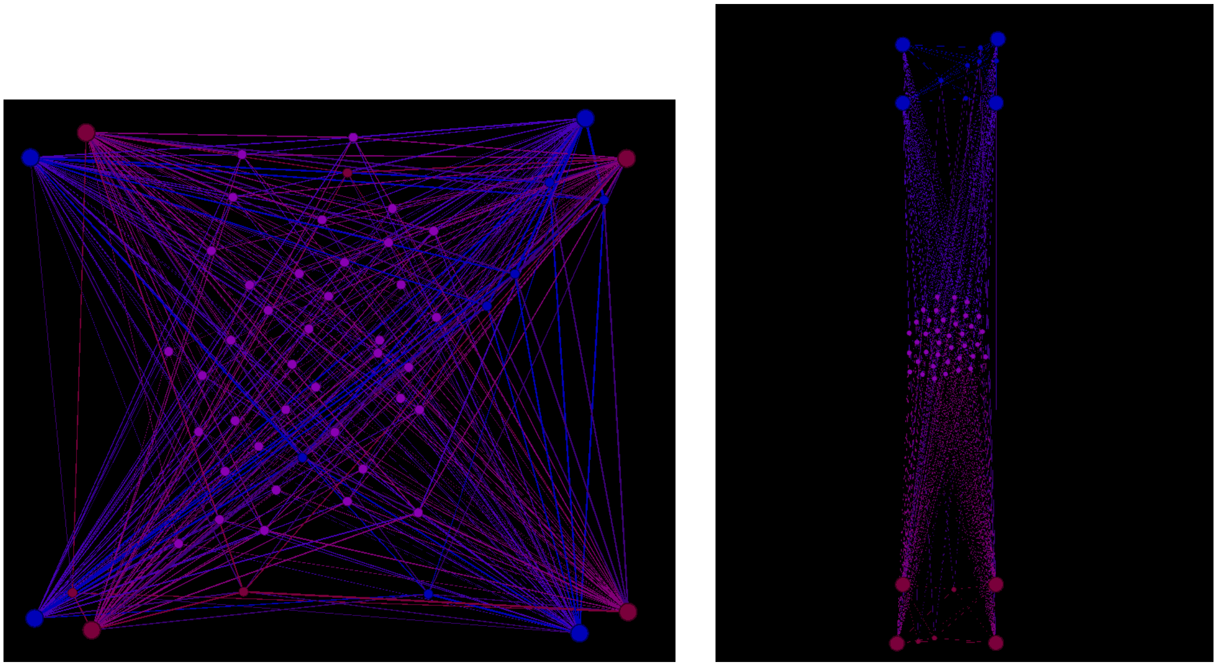


Image 4.14: spatial configuration of the nodes in a) two and b) three dimensions

This is the spatial configuration of the nodes in two and three dimensions. The larger, corner nodes represent eight networks. We can see that some topics are more prominent in the “winner” networks (upper portion of the image), some in the “loser” networks (lower portion of the image), while most are in the middle. The same can be said about the time dimension. Similarly, there are three topics strongly affiliated with republicans and six with the democrats. Again, most topics are in between, colored purple.

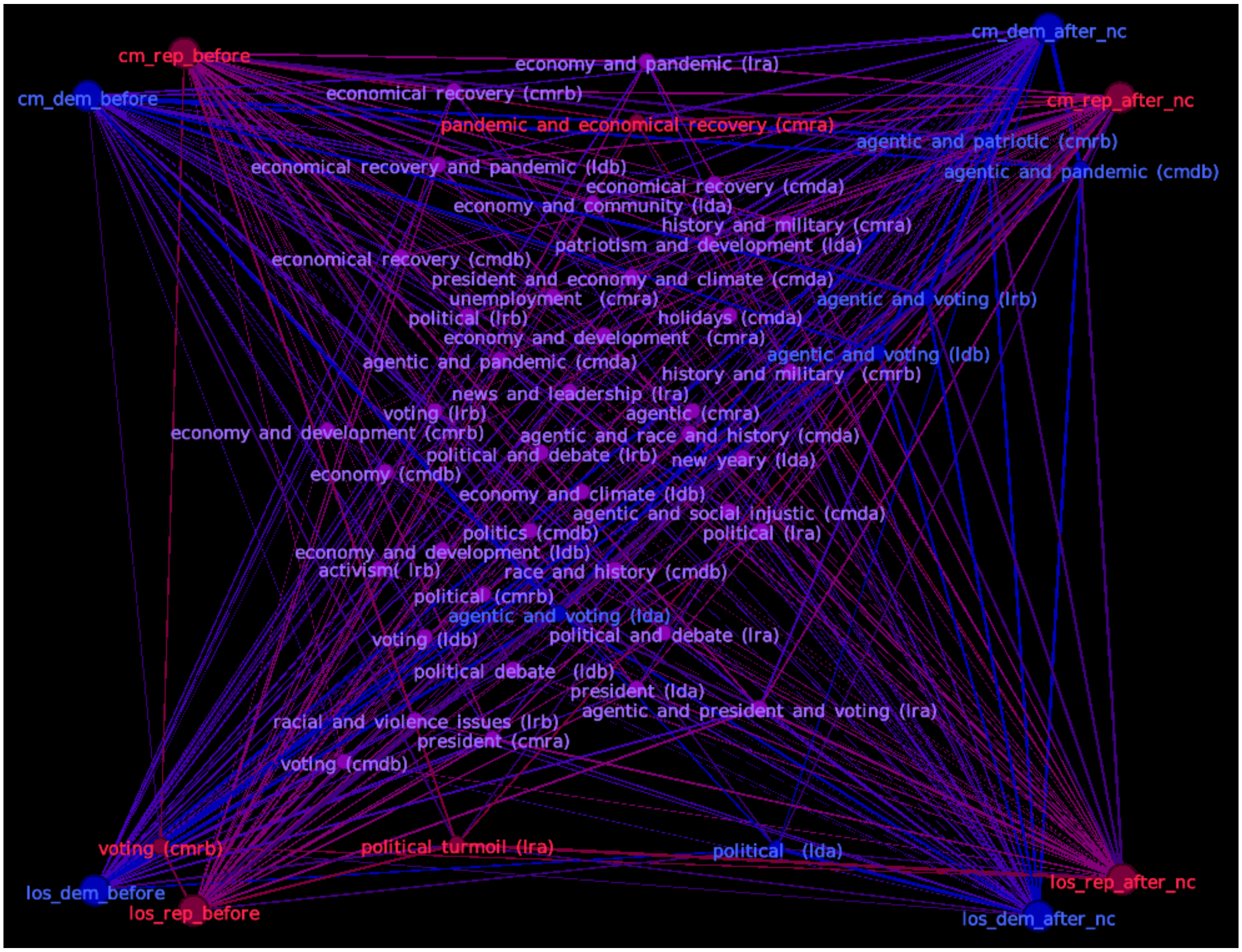

cm_dem_after_nc
cm_rep_before
economy and pandemic (lra)
cm_dem_before
economical recovery (cmrb)
cm_rep_after_nc
pandemic and economical recovery (cmra)
agentic and patriotic (cmrb)
economical recovery and pandemic (ldb)
agentic and pandemic (cmdb)
economical recovery (cmda)
economy and community (lda)
history and military (cmra)
patriotism and development (lda)
economical recovery (cmdb)
president and economy and climate (cmda)
unemployment (cmra)
agentic and voting (lrb)
political (lrb)
holidays (cmda)
economy and development (cmra)
agentic and voting (ldb)
agentic and pandemic (cmda)
history and military (cmrb)
news and leadership (lra)
voting (lrb)
agentic (cmra)
economy and development (cmrb)
agentic and race and history (cmda)
political and debate (lrb)
new yeary (lda)
economy (cmdb)
economy and climate (ldb)
agentic and social injustic (cmda)
politics (cmdb)
political (lra)
economy and development (ldb)
activism(lrb)
race and history (cmdb)
political (cmrb)
agentic and voting (lda)
voting (ldb)
political and debate (lra)
political debate (ldb)
president (lda)
agentic and president and voting (lra)
racial and violence issues (lrb)
president (cmra)
voting (cmdb)
voting (cmrb)
political turmoil (lra)
political (lda)
los_rep_after_nc
los_dem_before
los_rep_before
los_dem_after_nc

Image 4.15: spatial configuration of the nodes with labels

We can see that topics having to do with the economy are more typical of the “before” networks (scattered around the left hand side and the middle of the image), which correlates to the typical pre-election promises politicians tend to make around the election. Agentic topics on the other hand are more prominent in the “after” networks, with few exceptions that show up in the middle but those topics are mixed with voting and race. The topics in the aftermath of the elections became more political and centered around the president, debate, social injustice, which we believe reflects the post-election turmoil and the debate on the validity of the presidential election.

Topics typical for the winners are talking about the pandemic and economical recovery, are agentic, patriotic and evoking history. Topics closer to the loser networks are in relation to voting, political turmoil, racial and violence issues and the president.

The three republican topics are about voting, political turmoil and economic recovery and pandemic response. The typical democratic topics are agentic together with voting, patriotism and pandemic as well as one political topic (which is affiliated with the loser networks and post election period).

4.6 Conclusion

We identified communities in eight different networks. The six most prominent communities by node number and PageRank value of their nodes talked about the same topics across the eight networks, albeit in slightly different ways. We calculated the correlation between the 48 communities and found correlation across different networks. To capture this in a clearer way, we then named the topics and calculated their PageRank in each of the networks, as a sum of the PageRank of the topics' nodes. We used these values as distance between the topics and the networks. Visualizing this in three-dimensional space helped us show how the discourse changes over time (before and after elections), which discourse was typical for the successful versus unsuccessful candidates and which topics are more affiliated with which political party. We can see that the successful candidates used agentic words together with other topics (such as voting, pandemic, patriotism) as well as the economy and economic recovery in response to the pandemic. We also saw that the topics pertaining to the economy and unemployment were more typical before the election. Finally, we saw that the topic of political issues, debate and turmoil was more likely to appear in the tweets posted by the unsuccessful candidates. Finally, the agentic topics were closer to the democrats' networks.

5. Summary and conclusion

Twitter data offers various possibilities for interpretation within the context of network science. The focus of this thesis was to study a special kind of networks, the semantic networks, built from the tweet text, hashtags and metadata. More specifically, we aimed at detecting meaningful communities based on the topics in order to study the language used within the context of those topics. These results will be a basis for further study of the data.

Although in both projects tweets are used to study language on social media and its psycho-social implications, the main goal differs and so does the methodology applied to community detection.

In the first study, Comparing the role of men within prolife and prochoice community in discussion about abortion on Twitter, the aim was to detect a predetermined topic inside the bigger scope of discussion. The data collection was keyword based and any information we have about the users who posted the tweets is implicit in the results, and not a priori knowledge, save for their assumed prolife or prochoice stance on the abortion. It was a challenge to find a modality of data collection which would give us a substantial amount of data to study, within the desired time frame. We looked at tweets posted in 2019 because that is the year when the complimentary study (not described here) was done. The dataset was comparably small in relation to the existing data on the topic of abortion on Twitter. Another challenge was separating prolife and prochoice data. Even though our presumption that they would contain the words "prolife" and "prochoice", respectively, that is not always the case. We had to manually inspect the tweets which gained prominence in the dataset in order to discard the wrongly attributed tweets. The usual way to build a network from tweet data is connecting the words and hashtags. The results of community detection in such a network were not meaningful for the goal of our discussion, but they did validate our dataset by showing clear distinction between prolife and prochoice and by having communities that corresponded to the important events, around which the tweeting activity was high in the both datasets. We turned to the novel concept of "annotations", a metadata attached to tweets starting from Twitter API 2.0. Even though it left us with a smaller dataset, as not all tweets are annotated, the results of the community detection were meaningful and most importantly, showed that the topic of fatherhood is present and prominent in both datasets. When comparing nodes that belong to the "fatherhood" community in prolife and prochoice networks the conflicting sentiment between the two groups was evident. Moreover, by comparing PageRanks of the nodes, we showed that the topic of fatherhood is slightly more prominent in the prolife dataset, as were the male-related keywords, with the exception of "men". However, we showed that the use of the word "men" is ambiguous since it is often used to denote gender and not necessarily a male parent. Another ambiguous word was "father", which was used in the prolife network to denote god. The size of the dataset, the absence of prominent hashtags which would promote the inclusion of men in the discussion as well as the context in which the topic of fatherhood was placed all point to the conclusion that the discussion on the role of men is marginal, albeit slightly more prominent in the prolife group and that both groups only include men in the discussion if their stance on the abortion is corresponding to the group's.

In the second study, Analyzing political discourse before and after election using Tweets posted by the candidates for the 2020 US Elections we targeted a specific set of users and tried to identify topics they use as well as the difference in the topics over time, party lines and between the successful and unsuccessful candidates. Unlike in the previous study, our aim was not to detect a specific topic, but to discover the topics from the tweets. We also had information on the users which helped us differentiate between the successful and unsuccessful candidates and between democrats and republicans. Data collection was challenging because it consisted of multiple steps that had to be validated. First, we had to find a way to programmatically isolate the names of candidates (both successful and not). Then, we had to collect their Twitter profiles, which proved especially hard for the unsuccessful candidates since they were not that likely to be affiliated with senate/congress based keywords. Many of them also deleted their profiles. We had to manually inspect the datasets to discard bogus profiles which were picked up by the search. Since we had a lot of unsuccessful candidates missing we also performed manual search for their profiles, cross-referencing the results with Ballotpedia.
We found that the most meaningful community detection is observed when we study the word projection of a bipartite word-hashtag network. We created eight such networks for eight different scenarios we were interested in. The dataset was substantial, especially for the chosen members dataset. The word projection networks we created had to be constricted, otherwise working with them was computationally infeasible due to a large number of edges. Community visualization showed that the communities do not differ that much across different scenarios and that similar topics appear in the six biggest communities. This was further confirmed by calculating and visualizing the community correlation matrix. However, this visualization did not offer a very clear representation of what we wanted to show in the network. The next step was to name the communities in each network and then compare them across the eight networks to see how prominent words from the topics are in the network and assign a network similarity measure to each topic. We then created a network of topics and networks and using a specialized layout algorithm placed the networks as reference points in space with nodes being close to whatever network they are most similar to based on time (before vs after) and outcome (winner vs losers) scores. We also projected the network in three dimensions with the NetworkSplitter3D algorithm, using the "party" score as the third dimension. Since this was hard to visualize with labels, we used the results of this algorithm to color the nodes. Most nodes were placed in the middle of the network. However, there were topics that clearly and coherently belong to either side of the temporal dimension and the outcome dimension. There were three communities more affiliated with the republicans and six communities more affiliated with democrats.

The differences between the two studies show that there is no single modality that can be applied to different topics. Each carries its own peculiarities and has to be approached from different angles, until the best results are achieved. In order to get meaningful results we need a substantial dataset, proper methods of cleaning and validating the data and the right approach to network building. The discussion of the results also differs and while sometimes meaningful results are immediately found with community detection and PageRank, this often has to be supplemented with innovative ways of connecting and visualizing results.

In both studies we achieved the initial goal. The results we achieved will be used to further enrich the study of the project topic. Methods such as semantic analysis of the tweets, assigning tweets to communities by using the PRLP method, similarity of users to topics and individual user temporal analysis will be conducted. Some of these analyses move the discussion away from the pure network science and are out of scope of this thesis. Finally, to properly interpret the result an interdisciplinary approach is needed. The results of this thesis will be studied in the context of linguistics and social psychology.

6. References

[1] Network Science, A.-L. Barabási [Online]. http://networksciencebook.com/

[2] The PageRank Citation Ranking: Bringing Order to the Web (1998), Larry Page, Sergey Brin, R. Motwani, T. Winograd

[3] Community Detection Algorithms: A Comparative Analysis (2009), Lancichinetti and Fortunato

[4] Ballotpedia, https://ballotpedia.org/Main_Page. Retrieved on October 20, 2021.

[5] A simple example of Multidimensional Scaling with R and Gephi, Wouter Spekkink, https://www.wouterspekkink.org/r/mds/gephi/2015/12/15/a-simple-example-of-mds-using-r-and-gephi.html. Retrieved on January 18, 2022.

[6] *The Gephi Network Splitter 3D Layout,* 2014, Alexandre Barão. http://www.relationalcapitalvalue.com/gephiplugins.html. Retrieved on January 18, 2022.

Code appendix

1.Authorization header creation

```
import os
import requests
import time

os.environ['TOKEN'] = ''
bearer_token = os.environ['TOKEN']

def create_headers(bearer_token):
   headers = {"Authorization": "Bearer {}".format(bearer_token)}
   return headers

headers = create_headers(os.environ['TOKEN'])
```

2. Twitter API data collection

2.1 Counts endpoint

```
def create_url_counts(keyword, start_date, end_date, endpoint):
 search_url = endpoint
 query_params = {'query': keyword,
                 'start_time': start_date,
                 'end_time': end_date,
 }
 return (search_url, query_params)
```

```
def connect_to_endpoint_counts(url, headers, params, next_token = None):
   print(next_token)
   if next_token is not None and next_token != '':
     params['next_token'] = next_token
   response = requests.request("GET", url, headers = headers, params = 
params)
   if response.status_code != 200:
       raise Exception(response.status_code, response.text)
   return response.json()

def get_data_counts(keyword, start_time, end_time, next_token, endpoint):
 results = 0
 result_count = {}
  while next_token is not None:
   try:
     url = create_url_counts(keyword, start_time, end_time, endpoint)
     json_response = connect_to_endpoint_counts(url[0], headers, url[1], 
next_token)
     if "data" in json_response:
       for date in json_response["data"]:
         result_count[date["start"]] = date["tweet_count"]
     if "meta" in json_response:
       if "next_token" in json_response["meta"].keys():
         next_token = json_response["meta"]["next_token"]
       else:
         next_token = None
       if "total_tweet_count" in json_response["meta"].keys():
         results += int(json_response["meta"]["total_tweet_count"])
     else:
       next_token = None
   except Exception as e:
     print("Error occurred", e)
 print("Done")

 return (results, next_token, result_count)
```

2.2 Search tweets endpoint

```
def create_url(keyword, start_date, end_date, endpoint):

   search_url = endpoint
   query_params = {'query': keyword,
                   'max_results': 100,
                   'start_time': start_date,
                   'end_time': end_date,
                   'expansions':
'author_id,in_reply_to_user_id,geo.place_id',
                   'tweet.fields':
'id,text,author_id,in_reply_to_user_id,geo,conversation_id,created_at,lang
,public_metrics,referenced_tweets,reply_settings,source,context_annotation
s,entities',
                   'user.fields':
'id,name,username,created_at,description,public_metrics,verified',
                   'place.fields':
'full_name,id,country,country_code,geo,name,place_type',
                   'pagination_token': {}}
   return (search_url, query_params)

def connect_to_endpoint(url, headers, params, next_token = None):
   print(next_token)
   if next_token is not None and next_token != '':
     params['next_token'] = next_token
   response = requests.request("GET", url, headers = headers, params =
params)
   if response.status_code != 200:
       raise Exception(response.status_code, response.text)
   return response.json()

def get_data(keyword, start_time, end_time, next_token, endpoint, path):
 results = []
  while next_token is not None:
   try:
     url = create_url(keyword, start_time, end_time, endpoint)
     json_response = connect_to_endpoint(url[0], headers, url[1],
next_token)
     if "data" in json_response:
       results.extend(json_response["data"])
     if "meta" in json_response:
       if "next_token" in json_response["meta"].keys():
         next_token = json_response["meta"]["next_token"]
       else:
         next_token = None
     else:
       next_token = None
     time.sleep(3)
   except Exception as e:
     print("Error occurred", e)
 print("Done")

 return (results, next_token)
```

2.3 Search users endpoint

```
def create_url(keyword):

   search_url = "https://api.twitter.com/1.1/users/search.json" #Change to
the endpoint you want to collect data from

   query_params = {'q': keyword}
   return (search_url, query_params)

def connect_to_endpoint(url, headers, params):
   response = requests.request("GET", url, params=params, headers =
headers)
   if response.status_code != 200:
       raise Exception(response.status_code, response.text)
   return response.json()
```

2.3.1 Winners

```
result_df = pd.DataFrame()

for idx, row in chosen_members.iterrows():
 keyword = row["Member"] + " " + row["Type"]
 url = create_url(keyword)
 json_response = connect_to_endpoint(url[0], headers, url[1])

 if bool(json_response) == False:
   counter = 0
   roles = ["Congressman", "Congresswoman", "Congress"]

   for role in roles:
     keyword = row["Member"] + " " + role
     url = create_url(keyword)
     json_response = connect_to_endpoint(url[0], headers, url[1])
     if bool(json_response):
       break
  for result in json_response:
   print(keyword)
   user = pd.Series.to_frame(pd.DataFrame.from_dict(result,
orient='index')[0][["id_str","screen_name","created_at","statuses_count"]]
).T
   user["Member"] = keyword
   result_df = result_df.append(user)
  #time.sleep(seconds)

result_df = result_df.reset_index().drop(columns=['index'])
```

2.3.2 Losers

```
result_df = pd.DataFrame()

for idx, row in chosen_members.iterrows():
 keyword = row["Member"] + " " + row["Type"]
 url = create_url(keyword)
 json_response = connect_to_endpoint(url[0], headers, url[1])
 time.sleep(2)

 if bool(json_response) == False:
   counter = 0
   roles = ["Congressman", "Congresswoman", "Congress"]

   for role in roles:
     keyword = row["Member"] + " " + role
     url = create_url(keyword)
     json_response = connect_to_endpoint(url[0], headers, url[1])
     time.sleep(2)
     if bool(json_response):
       break
  for result in json_response:
   user = pd.Series.to_frame(pd.DataFrame.from_dict(result,
orient='index')[0][["id_str","screen_name","created_at","statuses_count"]]
).T
   user["Member"] = keyword
   result_df = result_df.append(user)
  print(user["screen_name"])

result_df = result_df.reset_index().drop(columns=['index'])
```

2.4 Users timeline endpoint

```
def create_url(id, start_date, end_date):

   search_url = "https://api.twitter.com/2/users/"+id+"/tweets" #Change to
the endpoint you want to collect data from

   #change params based on the endpoint you are using
   query_params = {
                   'start_time': start_date,
                   'end_time': end_date,
                   'expansions':
'author_id,in_reply_to_user_id,geo.place_id',
                   'tweet.fields':
'id,text,author_id,in_reply_to_user_id,geo,conversation_id,created_at,lang
,public_metrics,referenced_tweets,reply_settings,source,context_annotation
s,entities',
                   'user.fields':
'id,name,username,created_at,description,public_metrics,verified',
                   'place.fields':
'full_name,id,country,country_code,geo,name,place_type',
                   'pagination_token': {}}
   return (search_url, query_params)
def connect_to_endpoint(url, headers, params, next_token = None):
   params['pagination_token'] = next_token   #params object received from
create_url function
   response = requests.request("GET", url, headers = headers, params =
params)
```

```
   #print("Endpoint Response Code: " + str(response.status_code))
   if response.status_code != 200:
       raise Exception(response.status_code, response.text)
   return response.json()

start_time = "2020-02-09T00:00:00.000Z"
end_time = "2021-08-09T00:00:00.000Z"

json_response = {}
json_response["meta"] = {}
json_response["meta"]["next_token"] = {}
next_token = {}
tweets = []

for idx, row in chosen_members_twitter.iterrows():
 print(row["screen_name"])
  user_timeline = []
 while next_token is not None:
   url = create_url(str(row["id_str"]), start_time,end_time)
   json_response = connect_to_endpoint(url[0], headers, url[1],
next_token)
   if "data" in json_response:
     user_timeline.extend(json_response["data"])
     tweets.extend(json_response["data"])
   if "meta" in json_response:
     if "next_token" in json_response["meta"]:
       next_token = json_response["meta"]["next_token"]
     else:
       next_token = None
   else:
     next_token = None
   time.sleep(1)

pd.DataFrame(user_timeline).to_csv(path, index=False)
```

3. Wikipedia scraping

```
from bs4 import BeautifulSoup
```

```
url_reps =
"https://en.wikipedia.org/wiki/2020_United_States_House_of_Representatives
_elections"
url_senators =
"https://en.wikipedia.org/wiki/2020_United_States_Senate_elections#Seats"
```

3.1 Winners

```
def get_senators(url):
 response = requests.get(url=url)
 soup = BeautifulSoup(response.content, 'html.parser')

 tag = soup.findAll('b')
 chosen_members = {}
 for elem in tag:
   tick = elem.find(alt='Green tick')
   if tick:
     if str.split(elem.find_previous_sibling().get("style"),'#')[1] ==
"3333FF":
       party = "Democratic"
     else:
       party = "Republican"
     chosen_members[str.strip(tick.parent.text[2:])] = party
  return chosen_members

def get_representatives(url):
 response = requests.get(url=url,)
 soup = BeautifulSoup(response.content, 'html.parser')
  counter = 0
 tag = soup.findAll('li')
 chosen_members = {}
 for elem in tag:
   tick = elem.find(alt='Green tick')
   if tick:
     counter += 1
     #exclude special elections
     if counter>5:
       text = str.split(elem.text,'(')
       chosen_members[str.strip(text[0][2:])] = str.split(text[1],')')[0]
 return chosen_members

chosen_senators = get_senators(url_senators)
chosen_representatives = get_representatives(url_reps)

non_voting = ["Amata Coleman Radewagen","Eleanor Holmes Norton","Michael
San Nicolas","Gregorio Kilili Sablan","Jenniffer González","Stacey
Plaskett"]

for nv_member in non_voting:
 chosen_representatives.pop(nv_member)

chosen_senators_df = pd.DataFrame.from_dict(chosen_senators,
orient="index").reset_index().rename(columns={'index':'Member',0:'Party'})
chosen_senators_df["Type"] = "Senator"

chosen_reps_df = pd.DataFrame.from_dict(chosen_representatives,
orient="index").reset_index().rename(columns={'index':'Member',0:'Party'})
chosen_reps_df["Type"] = "Representative"
chosen_members = chosen_senators_df.append(chosen_reps_df)
chosen_members = chosen_members.reset_index().drop(columns=["index"])
```

3.2 Losers

```
def get_loser_representatives(url):
 response = requests.get(url=url,)
 soup = BeautifulSoup(response.content, 'html.parser')
  counter = 0
 tag = soup.findAll('li')
 chosen_members = {}
 for elem in tag:
   tick = elem.find(alt='Green tick')
   if tick:
     counter += 1
     #exclude special elections
     if counter>5:
       loser = tick.find_parent().find_next_siblings()
       if loser:
         if isinstance(loser, list):
           for person in loser:
             text = str.split(person.text,'(')
             chosen_members[str.strip(text[0][0:])] =
str.split(text[1],')')[0]
         else:
           text = str.split(loser.text,'(')
           chosen_members[str.strip(text[0][0:])] =
str.split(text[1],')')[0]

 return chosen_members

def get_loser_senators(url):
 response = requests.get(url=url,)
 soup = BeautifulSoup(response.content, 'html.parser')

 tag = soup.findAll('li')
 chosen_members = {}
 for elem in tag:
   tick = elem.find(alt='Green tick')
   if tick:
     loser = tick.find_parent().find_parent().find_next_siblings()
     if isinstance(loser, list):
       for person in loser:
         text = str.split(person.text,"(")
         try:
           chosen_members[str.strip(text[0][1:])] =
str.split(text[1],")")[0]
         except IndexError:
           print("Skipping...", text)
     else:
       print(text)
  return chosen_members

loser_representatives = get_loser_representatives(url_reps)
loser_senators = get_loser_senators(url_senators)

#delete non-voting
non_voting = ["Oreta Tufuga Mapu Crichton","Meleagi
Suitonu-Chapman","Patrick Hynes","Barbara Washington Franklin","Omari
Musa","Natale Lino Stracuzzi","Amir Lowery","David Krucoff","John
Cheeks","Robert Underwood"]

for nv_member in non_voting:
 loser_representatives.pop(nv_member)

loser_senators_df = pd.DataFrame.from_dict(loser_senators,
orient="index").reset_index().rename(columns={'index':'Member',0:'Party'})
loser_senators_df["Type"] = "Senator"

loser_reps_df = pd.DataFrame.from_dict(loser_representatives,
orient="index").reset_index().rename(columns={'index':'Member',0:'Party'})
loser_reps_df["Type"] = "Representative"

chosen_members = loser_senators_df.append(loser_reps_df)
chosen_members = chosen_members.reset_index().drop(columns=["index"])
```

4. Text cleaning

```
# NLTK tools
import nltk
nltk.download('words')
words = set(nltk.corpus.words.words())
words.add('prolife')
words.add('prochoice')
words.add('boyfriend')
nltk.download('stopwords')
#removed possessive pronouns my, your, yours, our/s, their/s, hers, his
stop_words = {'a', 'about', 'above', 'after', 'again', 'against', 'ain',
'all', 'am', 'an', 'and', 'any', 'are', 'aren', "aren't", 'as', 'at',
'be', 'because', 'been', 'before', 'being', 'below', 'between', 'both',
'but', 'by', 'can', 'couldn', "couldn't", 'd', 'did', 'didn', "didn't",
'do', 'does', 'doesn', "doesn't", 'doing', 'don', "don't", 'down',
'during', 'each', 'few', 'for', 'from', 'further', 'had', 'hadn',
"hadn't", 'has', 'hasn', "hasn't", 'have', 'haven', "haven't", 'having',
'he', 'her', 'here',  'herself', 'him',  'himself',  'how', 'i', 'if',
'in', 'into', 'is', 'isn', "isn't", 'it', "it's", 'its', 'itself', 'just',
'll', 'm', 'ma', 'me', 'mightn', "mightn't", 'more', 'most', 'mustn',
"mustn't", 'myself', 'needn', "needn't", 'no', 'nor', 'not', 'now', 'o',
'of', 'off', 'on', 'once', 'only', 'or', 'other', 'ourselves', 'out',
'over', 'own', 're', 's', 'same', 'shan', "shan't", 'she', "she's",
'should', "should've", 'shouldn', "shouldn't", 'so', 'some', 'such', 't',
'than', 'that', "that'll", 'the', 'them', 'themselves', 'then', 'there',
'these', 'they', 'this', 'those', 'through', 'to', 'too', 'under',
'until', 'up', 've', 'very', 'was', 'wasn', "wasn't", 'we', 'were',
'weren', "weren't", 'what', 'when', 'where', 'which', 'while', 'who',
'whom', 'why', 'will', 'with', 'won', "won't", 'wouldn', "wouldn't", 'y',
'you', "you'd", "you'll", "you're", "you've", 'yourself', 'yourselves'}
nltk.download('wordnet')
from nltk.corpus import wordnet as wn
from nltk.stem.wordnet import WordNetLemmatizer
from nltk import pos_tag
nltk.download('averaged_perceptron_tagger')
from collections import defaultdict
tag_map = defaultdict(lambda : wn.NOUN)
tag_map['J'] = wn.ADJ
tag_map['V'] = wn.VERB
tag_map['R'] = wn.ADV


def cleaner(tweet):
   tweet = re.sub("@[A-Za-z0-9]+","",tweet) # remove mentions
   tweet = re.sub("#[A-Za-z0-9]+", "",tweet) # remove hashtags
   tweet = re.sub(r"(?:\@|http?\://|https?\://|www)\S+", "", tweet) #
remove http links
   tweet = " ".join(tweet.split())
   tweet = " ".join(w for w in nltk.wordpunct_tokenize(tweet) \
        if w.lower() in words and not w.lower() in stop_words) #remove
stop words
   lemma_function = WordNetLemmatizer()
   tweet = " ".join(lemma_function.lemmatize(token, tag_map[tag[0]]) for
token, tag in nltk.pos_tag(nltk.wordpunct_tokenize(tweet))) #lemmatize
   tweet = str.lower(tweet) #to lowercase
   return tweet
```

5. Create network graph

```
#to get weighted graph we need a list of 3-element tuples (u,v,w) where u
and v are nodes and w is a number representing weight
up_weighted = []
for edge in network:
   #we can filter edges by weight by uncommenting the next line and
setting desired weight threshold
   #if(network[edge])>1:
   up_weighted.append((edge[0],edge[1],network[edge]))

G = nx.Graph()
G.add_weighted_edges_from(up_weighted)

#for hashtag networks
G.remove_node("#"+keyword)

for cc in nx.connected_components(G):
 print(cc)

S = [G.subgraph(c).copy() for c in nx.connected_components(G)]
G  = S[0]
```

5.1 Color the nodes

5.1.1 Topic network

```
color = {}
for node in G.nodes:
 if node[0].isupper() or node[0].isdigit():
   color[node] = 1
 else:
   color[node]  = 0

nx.set_node_attributes(G, color, "bipartite")
```

5.1.2 Hashtag network

```
color = {}
for node in G.nodes:
 if node.startswith("#"):
   color[node] = 1
 else:
   color[node]  = 0

nx.set_node_attributes(G, color, "bipartite")
```

5.2 Create word projection

```
print(bipartite.is_bipartite(G))

words = {n for n, d in G.nodes(data=True) if d["bipartite"] == 0}

hashtags = set(G) - words

word_projection = bipartite.weighted_projected_graph(G, words)
```

6. Community correlation

```
dict_of_comms = {}
dict_of_nodes = {}

filenames = ["cm_dem_after_nc.csv",
            "cm_dem_before.csv",
            "cm_rep_after_nc.csv",
            "cm_rep_before.csv",
            "los_dem_after_nc.csv",
            "los_dem_before.csv",
            "los_rep_after_nc.csv",
            "los_rep_before.csv"]

for filename in filenames:
 print(filename)
 nodes_df = pd.read_csv(filename)
 nodes_df = nodes_df[["Id", "pageranks", "modularity_class"]]#,
"filter_pr_comm"]]
```

```
 nodes_df = nodes_df.rename(columns={"Id":"id"})
 nodes_df.sort_values(by="modularity_class")

 communities_df =
nodes_df.groupby('modularity_class').size().reset_index(name='community_si
ze').sort_values(by='community_size', ascending=False)
 num_nodes = len(nodes_df)
 communities_df["network_percentage"] =
round(communities_df["community_size"]/num_nodes*100,2)
 communities_df =
communities_df.merge(nodes_df.groupby(['modularity_class'])['pageranks'].a
gg('sum').reset_index(), on="modularity_class")

 map = {}
 counter = 1

 for idx, row in communities_df.iterrows():
   map[row["modularity_class"]] = counter
   counter += 1
  communities_df["new_class"] = communities_df["modularity_class"]
 communities_df["new_class"] =
communities_df["modularity_class"].map(lambda x: map[x])

 nodes_df = nodes_df.merge(communities_df, on="modularity_class")
 #nodes_df = nodes_df[nodes_df["filter_pr_comm"]==True]
 nodes_df = nodes_df.drop(columns=["modularity_class"])
 nodes_df = nodes_df.rename(columns={"new_class":"modularity_class",
"pageranks_x":"pageranks"})
 nodes_df = nodes_df.sort_values(by="modularity_class")
 nodes_df = nodes_df[["id", "pageranks", "modularity_class"]]
 nodes_df = nodes_df.reset_index().drop(columns=["index"])

 communities_df = communities_df[0:6]
 communities_df =communities_df.drop(columns=["modularity_class"])
 communities_df =
communities_df.rename(columns={"new_class":"modularity_class"})

 source = filename.split(".csv")[0]
 communities_df["source"] = source
 nodes_df["source"] = source
```

```
  dict_of_comms[source] = communities_df
 dict_of_nodes[source] = nodes_df

keys = set(dict_of_comms.keys())
dict_of_intersections = {}

for a in keys:
  community_matrix_a = dict_of_comms[a]
 comm_a_list = list(community_matrix_a["modularity_class"])

 for comm_a in comm_a_list:
   dict_of_intersections[a+"_"+str(comm_a)] = {}
  for b in keys:
   community_matrix_b = dict_of_comms[b]
   comm_b_list = list(community_matrix_b["modularity_class"])

   for comm_a in comm_a_list:
     for comm_b in comm_b_list:
       dict_of_intersections[a+"_"+str(comm_a)][b+"_"+str(comm_b)] = {}

i = 0
for key_a, community_a in dict_of_intersections.items():
 for key_b, community_b in community_a.items():

   community_a_number = key_a[-1]
   community_b_number = key_b[-1]

   network_a = key_a.split("_"+community_a_number)[0]
   network_b = key_b.split("_"+community_b_number)[0]

   nodes_a = dict_of_nodes[network_a]
   nodes_a = nodes_a[nodes_a["modularity_class"]==int(community_a_number)]

   nodes_b = dict_of_nodes[network_b]
   nodes_b = nodes_b[nodes_b["modularity_class"]==int(community_b_number)]


   intersection = pd.merge(nodes_a, nodes_b, on="id",
suffixes=('_'+key_a,'_'+key_b))
```

```
   if key_a == key_b:
     intersection = intersection["pageranks_"+key_a].sum()[0]
   else:
     intersection = intersection["pageranks_"+key_a].sum()

   dict_of_intersections[key_a][key_b] = intersection


correlation_matrix = pd.DataFrame.from_dict(dict_of_intersections,
orient="index")
```

7. Similarity of topic to network

```
dict_of_topics = {}

for key, value in dict_of_nodes.items():
 value = value[value["modularity_class"]<7]
 for i in range(1,7):
   topic = key + "_" + str(i)

   dict_of_topics[topic] = value[value["modularity_class"]==i]

topic_rank_in_networks = {}

for topic_name, topic in dict_of_topics.items():
 topic_rank_in_networks[topic_name] = {}
 for network, nodes in dict_of_nodes.items():
   intersection = pd.merge(nodes, topic, on="id", suffixes=["_nodes",
"_topic"])
   topic_rank_in_networks[topic_name][network] =
intersection["pageranks_nodes"].sum()

 topic_rank_df = pd.DataFrame.from_dict(topic_rank_in_networks,
orient="index")

topic_rank_df["cm"] = topic_rank_df["cm_dem_after_nc"] +
topic_rank_df["cm_dem_before"] + topic_rank_df["cm_rep_after_nc"]  +
topic_rank_df["cm_rep_before"]
```

```
topic_rank_df["los"] =  topic_rank_df["los_dem_after_nc"] +
topic_rank_df["los_dem_before"] + topic_rank_df["los_rep_after_nc"]  +
topic_rank_df["los_rep_before"]
topic_rank_df["before"] =  topic_rank_df["cm_dem_before"] +
topic_rank_df["cm_rep_before"] + topic_rank_df["los_dem_before"]  +
topic_rank_df["los_rep_before"]
topic_rank_df["after"] =  topic_rank_df["cm_dem_after_nc"] +
topic_rank_df["cm_rep_after_nc"] + topic_rank_df["los_dem_after_nc"]  +
topic_rank_df["los_rep_after_nc"]
topic_rank_df["dem"] = topic_rank_df["cm_dem_after_nc"] +
topic_rank_df["cm_dem_before"] + topic_rank_df["los_dem_after_nc"] +
topic_rank_df["los_dem_before"]
topic_rank_df["rep"] = topic_rank_df["cm_rep_after_nc"] +
topic_rank_df["cm_rep_before"] + topic_rank_df["los_rep_after_nc"] +
topic_rank_df["los_rep_before"]

topic_rank_df["rep"] = - topic_rank_df["rep"]
topic_rank_df["los"] = - topic_rank_df["los"]
topic_rank_df["before"] = - topic_rank_df["before"]
topic_rank_df =
topic_rank_df.reset_index().rename(columns={"index":"Label"})

topic_rank_df = topic_rank_df[["Id", "Label","cm","los","before","after",
"dem", "rep"]]

topic_rank_df ["outcome"] = (topic_rank_df ["los"] + topic_rank_df ["cm"])
/2
topic_rank_df ["time"] = (topic_rank_df ["before"] + topic_rank_df
["after"]) /2
topic_rank_df ["party"] = (topic_rank_df ["rep"] + topic_rank_df ["dem"])
/2

topic_rank_df.to_csv("topic_rank_df.csv")
```