Individual Project

Graph database creation and Assortativity analysis in the context of online discourse

presented by
Alexander Haberling
Matriculation Number 1450868

submitted to the
Data and Web Science Group
Prof. Dr. Stuckenschmidt
University of Mannheim

05. February 2021

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The goal of this project was for the author to acquire the skills necessary for conceptualizing, creating and populating a graph database. Furthermore, experience in applied network analysis was set to be gained. Additionally, these aspirations were pursued in the context of semantic data related to political online discourse.

In the following sections, this report demonstrates, how data from an mostly American, online debating website, a Neo4j graph database instance and the GraphTool Python library for network analysis were utilized to pursue these goals. After a brief introduction into the structure of the underlying data sets, the modulation, population and exportation of of the graph database is discussed. Following up on this, chapter 4 Preprocessing reveals how the resulting graph data was further preprocessed in order to facilitate an accurate, descriptive and assortativity focused analysis. The report is concluded by a brief summary of the achieved goals and a discussion of the prevailing limitations. An outlook into possible future efforts is granted as well.

Chapter 2

Data

In order to establish the before mentioned competencies, the data sets provided by Esin Durmus and Claire Cardie [7, 6] were utilized. The sets contain data scrapped from an online debating website named debate.org [1]. The website exist predominantly in an U.S. American context and features debates, opinions and polls about politics, morals and various other topics. In this chapter an insight into the structure of the website and the provided data sets is giving. Following, broad overall numbers of debate.org entities are presented to discuss the scope of the present data.

2.1 Structure of Discourse

In the context of debate.org a debate always involves two users arguing and an arbitrary number of users observing and commenting on the debate. Additionally users are able to vote on the performance of the participants in several categories including conduct, spelling and grammar, quality of arguments and sources (see Appendix B). A debate consists of a finite set of rounds. In each round, both participants present their arguments and possibly address the arguments of their opponent in one blog post each. At the end of a debate either one participant is crowned victor, or a tie is called based on the votes given by the judging users.

Debate.org opinions are allegedly controversial statements or questions presented in a simplified one liner. After an opinion was opened by an user, users have the opportunity to show their agreement or disagreement via a ”Yes” or ”No” vote. The structure of an opinion requires voters to support their vote with a publicly presented argument in text form. Their vote is publicly visible as well. A example for an opinion one liner, featured on the website would be: ”Do you agree with the Black Lives Matter Movement?”.

As last option of public discourse, user can initiate polls. A poll is similar in structure to an opinion. The main differences are that a supporting argument is not required and that voting categories are more customizable then ”Yes” or ”No” votes. An example featured on the website would be: ”Is god real?” with voting categories: ”God is real”, ”God isn’t real” and ”Don’t know if god is real”.

Debate.org also provides the typical structure of an online forum but its entries and existence are not accounted for in the provided data sets.

Apart from participating in the activities described above, users can enrich their profiles with demographic data and provide their stances regarding prominent political and moral issues such as abortion, drug legalization or gay marriage (see Appendix A). Stances are expressed in short terms including ”Pro”, ”Con” and ”Und”. The latter standing for undecided. Additionally, users can form online friendships in a Facebook-like manner. After a friend request is sent and accepted a reciprocal connection is establish. These two aspect of expressing stances and forming online friendships are utilized in the assortativity analysis latter on.

2.2 Data Set Features

The ”debates” data set [1] accounts for the scrapped debates, and comments and votes related to these debates. The ”users” data set [1] holds information about users in form of their above described demographics, political stances, opinion and poll participation and aggregates such as the number of won debates, the number of friends and the number of opinion contributions. Over all, the data sets feature 78.376 debates, 45.348 user profiles and 606.102 debate related comments. All debates together consist of 560.799 blog posts in which the two debating users exchange their arguments. Two blog posts (one from each participant) define a round in a debate. Additionally, all debates feature 398.412 vote maps given by users. One vote map contains the evaluation of one participant in a debate (see Appendix B). Following this, when rating debate performances, users always submit two vote maps per debate, one for each participant. Participating users are able to vote on their own debate performance, as well.

In the scraped time window 28.580 polls were issued and 288.649 poll votes were given in total, by all users. 24.209 opinions were opened with 60.449 user participations. Unfortunately the data sets provide only the title of the opinions and the comments accompanying the opinion votes. The information, whether the vote was ”Pro” or ”Con” is not present in the data sets. This information might be annotated by humans or inferred with semantical text analysis approaches, if needed.

In reference to the information provided by debates.org [2] the data sets contain a majority of the overall issued debates (91.93%), but only a fraction of the created user accounts (6,55%). However it is conceivable, that these 6,55% were the most active ones, in order to be captured by the scrapping process.

Both data sets are provided in Json format and are combined a little short of 1,5 GB. The ”debates” Json is indexed by a unique debate name, while the ”users” Json is indexed by a unique user name.

The authors state the window of data collection spans from October 2007 to November 2017. Unfortunately, 4.203 debates represented in the data set are created explicitly in 2018 (e.g. ”Zoos- Joys or Jails?”; start date ”4/27/2018”). Additionally, a specific date or time span of the data scrapping process is not provided. This leads to problems regarding relative time data. Some entities in the data set are explicitly dated in the mm/dd/yyyy format (e.g. start date of debates), others however are addressed relatively (e.g. ”created 6 months ago”). These anomalies are discussed in more detail in the following section 3.1.2 Time Dimension.

Chapter 3

Graph Database

The provided code for the creation and population of the graph database relies on a Neo4j Community Server instance versioned 4.1.1. For the exportation of the database, functions of the Neo4j APOC plug-in version 4.1.0.2 are utilized. All functionalities are accessed via the Neo4j Python Driver and implemented in Python 3.7. The code concerned with this database manipulation, as well as the code concerned with the upcoming analyses are publicly available on Github[3].

3.1 Conceptualization

A database structure suited to the underlying data and suited for upcoming projects and their expected queries enables fast and comfortable querying. The following database schema was not solely designed towards the upcoming assortativity analysis, but with a general queriability in mind. As an positive side effect, this lead to additional experience gathered in the context of graph manipulation via GraphTool, see section 5.2 Assortative Mixing

3.1.1 General Concept

Figure 3.1 represents the employed database schema. In order to facilitate an intuitive and functional usage, the central entities described in the section 2.1 Structure of Discourse, were employed as nodes. This lead to four node types named *User*, *Debate*, *Opinion* and *Poll*. Furthermore *Comment*, *VoteMap* and *Argument* where differentiated from the *Debate* nodes and established as their own node types. The political stances of users were differentiated as well, into another *Issues* type, leading to eight different node types. This way each *Issues* node corresponds to exactly one respective *User* node. The outlined fragmentation supports investigations focused on only a subset of the major aspects of the website. An illustration of this advantage would be a project concerned with the voting pattern of people that are virtually befriended. Following this schema information of vote maps could be accessed together with information about user profiles and their friendships, without having to handle the bulk of text information contained in *Argument* and *Comment* nodes. The *Argument* nodes store the blog posts that make up rounds detailed in section 2.1 Structure of Discourse. These two entities make up the biggest portion of data in terms of data size.

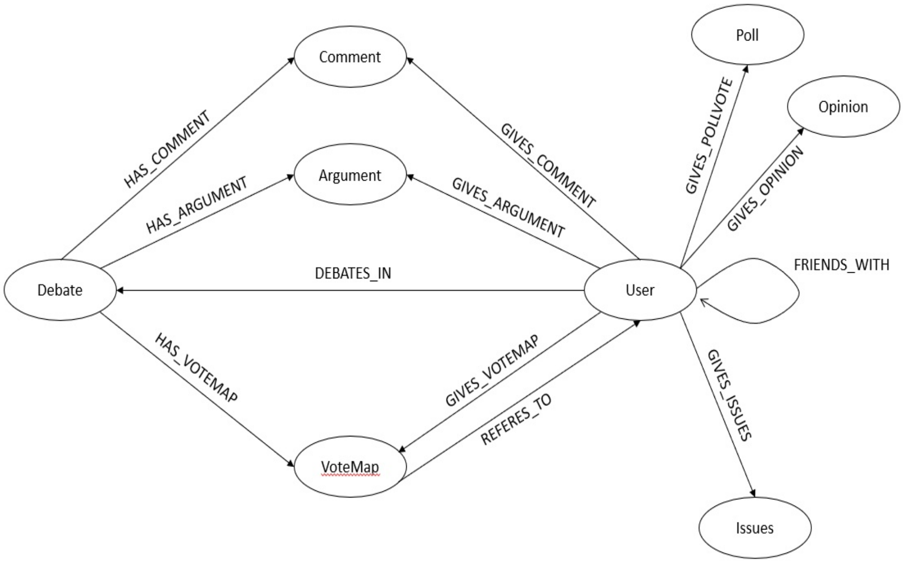

Figure 3.1: Graph Database Schema

When designing an appropriate structure for the database relations, *User* nodes are considered to be the focal entity of edges. *User* nodes are intuitively pointing to all other nodes. Apart from three exceptions, these pointing relations are labeled with the prefix GIVES_. A concrete example of this is the relation GIVES_VOTEMAP pointing from "User" nodes to *VoteMap* nodes. This is done to facilitate a more intuitive orientation in the database structure. This way users e.g. give comments, give arguments, give opinions and issues. Exceptions are the friendship relations FRIENDS_WITH between *User* nodes, the relations DEBATES_IN pointing from *User* nodes to *Debate* nodes and the IN_TIMELINE relations not depicted here and further discussed below. The first two of these exceptions might be renamed in future efforts, see section 6.1 Future Work & Limitations

The *Debate* nodes serve as secondary focal point to facilitate the other relations. Relations pointing from *Debate* nodes to other nodes are prefixed with HAS_ e.g. HAS_COMMENT. This way three additional edge types from *Debate* nodes to *Comment*, *Argument* and *VoteMap* nodes are established.

Last but not least *VoteMap* entities are not only connected via two incoming ties (GIVES_VOTEMAP, HAS_VOTEMAP) but also via an outgoing tie pointing towards *User* nodes. This outgoing tie is needed, since vote maps are not only given by users via voting, and not only contained in debates by their nature, but these vote maps also refer to a specific user. A user, being one of the participants of the debate. The possible critique of incorporating redundant information is addressed in subsection 3.1.3 Redundancy.

Most edges do not contain additional information beyond their source and target. Exceptions are the DEBATES_IN, GIVES_OPINION and GIVES_POLLVOTE containing data about debate performance and opinion and poll comments

3.1.2 Time Dimension

Most causal analyses rely on accounting for the aspect of time. For this purpose time concerned information relating to e.g. the date of debate creation, is modelled as well. Following [5] two common approaches of modelling the time aspect were investigated. The theoretically more favorable approach establishes ”before” or ”after” relations between entities of the same type. When laying out user profiles on a timescale regarding their date of creation, a profile would be connected with its direct predecessor/s and successor/s. In an optimal data structure each node would be connected via two time concerned relation with two other node, expect the first or last ever created node of the scrapped data. In theory this approach would support fast querying with minimal loss of information.

A second approach establishes nodes of type *Time*, *Year*, *Month* or more fine grained time spans. Each entity of the database appended with time related information is connected via an edge to the respective time nodes representing their time concerned variable (e.g. creation). Following this approach information is ”lost”, depending on how fine grained the time typed nodes are chosen to be. When considering two user profiles created on February 2011 and November 2011, and time dimension nodes grained on a year level, both *User* nodes would be connected to the node representing the year ”2011”. A differentiation of which node was created first is not supported in this case.

Contrasting to this, the former approach of ”before” (or ”after”) relations would clearly contain this information by either relating directly between these two nodes: user A ← user B, or by relating indirectly between these two nodes: user A ← ... ← user B.

During the population of the database it quickly became clear, that, despite its information loss, the second approach was more favorable. Reason for this is the ratio between the number of users in the data set and the number of most fine grained time units, e.g. days. When this ratio is skewed towards the number of nodes, then the number of before/after edges created between the nodes increases significantly, leading to a considerable longer period of database population and to a considerably increased cost of querying. An optimal case features at least one unit of time per nodes, leading to a 1:1 relation between nodes of neighboring time units. A worst case features an even distribution of nodes on two time units leading to n/2:n/2 relations (between nodes of neighboring time units). This problem is even aggravated for the relative time values (e.g. ”6 months ago”), since its translated time units are even more coarse grained.

Following the second approach, 13 additional nodes of type *Timeline* where added. One for each year in the span of 2007 to 2019. Edges of type IN_TIMELINE span from *User*, *Debate* and *Comment* nodes to their respective *Timeline* nodes. These nodes and edges have been omitted in Figure 3.1 for the sake of readability.

3.1.3 Redundancy

The final database arguably contains some redundant information. Examples of this are the node feature of user nodes concerned with the number of friends a user has, or the REFERS_TO relation. This data is saved an accessible in other ways as well (e.g. counting FRIENDS_WITH edges or traversing HAS_VOTEMAP edges). This is modeled by intention for the sake of comfortable and intuitive querying. The downsides in terms of database size are neglected with regard to the already manageable scale of the original Jsons and the resulting graph database.

3.2 Population

In order to populate a Neo4j graph database instance with the information contained in the debates.json and users.json data sets, the query language Cypher was utilized. Cypher was developed closely related to Neo4j and seems to be the standard language for manipulating data in Neo4j graph databases. This section refers to the debateDB_Creation.py provided via Github [3]

The data was extracted by parsing both Json files twice. This resulted in four major loops. In a first step both data sets where traversed to extract the relevant information for node creation. Subsequently both Json files where parsed again to extract edge relevant information. This differentiation deemed necessary, since edge creation relies on the source and target nodes to exist beforehand. Once both files are looped over the first time, meaning once all nodes are created, a Neo4j node index for each type of node is created. This indexing significantly reduces the time of the following edge creation. Additionally the differentiation into four loops supports systems with few computing power, by simplifying a partitioning of the code.

During each iteration, relevant information for the creation of the respective node or edge is extracted. After the extraction and still inside an iteration, the information is shipped as parameter of a function call. The functions called contain the respective ”MERGE”-queries that populate the database.

Inside the provided Python file a code structure is set, that facilitates the selection and exclusion of specific types of nodes and edges. This way the creation of databases containing not all information, but only subsets of, for a project relevant information is supported.

Code for both time concerned approaches described in section 3.1.2 Time Dimension is provided. Yet it is recommended to stick with the IN_TIMELINE approach. The Alternative approach has been commented out. For the IN_TIMELINE approach, relative time information (e.g. ”6 months ago”) was handled by inferring a hopefully accurate year. This was done by assuming an extraction time near the end of the official data scrapping window. Unfortunately the authors do not provide a date or time span of scraping. All relative time data was assigned to nodes of type Timeline by subtracting their value from 2017.

3.3 Exportation

The provided code in debateDB_Exportation.py yields two exemplary functions for a Neo4j database exportation. Both make use of the APOC library, a Neo4j lab project. The abbreviation stand for Awesome Procedures on Cypher and provides a comfortable exportation command, among other tings. One of the two described functions exports the database as a whole in Graphml format while the other one exports only *User* and *Issues* nodes and the edges between them, namely FRIENDS_WITH and GIVES_ISSUE. For the following preprocessing and analysis, the second, reduced exportation was used.

Chapter 4

Preprocessing

The preprocessing and all steps of the later presented analyses are conducted in Python 3.7 as well. In order to handle the graph structured nature of the data more easily, the GraphTool library is employed. GraphTools core data structures and algorithms are implemented in C++. This leads to benefits in form of performance, but comes with a cost of compatibility. Employing GraphTool on Windows is entangled with more difficulties then employing it on other popular operating systems. This was circumvented by accessing the library via a Docker image provided by its author Tiago P. Peixoto [4].

One caveat when working with Neo4j in combination with GraphTool is that a Graphml file exported via Neo4j is not automatically accessible in GraphTool. Due to a weird quirk of either Neo4j or the here presented employment of GraphTool, two lines have to be manually added to the exported Graphml file before importing it via GraphTool. The two lines are displayed in the appendix C as well as in the README.md accompanying this report and the provided code files[3].

Before a proper descriptive and assortativity focused analysis can be issued, the exported graph needs to be preprocessed. This is due to underlying issues with the FRIENDS_WITH relation.

The nature of debate.org online friendships between user profiles are similar to the nature of Facebook online friendships and many other online platforms. A friendship connection between two user is established iff one user sends a friendship request, that is afterwards accepted by its counterpart. This leads to debate.org friendships being inherently bidirectional. Unidirectional friendship relations are thereby, theoretically not supported in the data structure. Unfortunately these unidirectional friendships exist in the original Json data sets and thereby also in the exported Graphml file. A descriptive and assortativity focused analysis based on this Graphml file results in biased insights. In order to provide a more accurate presentations of the analysed friendship network, three preprocessing approaches are implemented and here presented.

Before diving into the different approaches, the investigated anomalies that make this preprocessing necessary are discussed. 94.376 friendships are featured in the original data sets. 45.135 of them are not bidirectional, but only unidirectional. Causes for this number are, on the one hand the option for debate.org users to enable private friendship settings. The Json value in the data set that is supposed to contain the list of befriended usernames, displays only a string value of ”private”, if this option is chosen. The scrapping of usable friendship data is thereby omitted in some instances.

Neo4j and GraphTool handle bidirectional relations as two unidirectional reciprocal relations. These two unidirectional relations are extracted from the Json by parsing over the two respective user nodes and by accessing their ”friends” value. In cases where one of the two users enabled private friendship settings during the scrapping period, only one of the two relations in created in den graph database. In cases where both user enabled the privacy setting, no edge is created and the friendship is undetected. Out of the 45.135 unsupported unidirectional edges, 44.804 are traced back to private friendship settings. The remaining 331 unjustified unidirectional FRIENDS_WITH edges are cause by 187 ”faulty” nodes. These nodes are nominated as friends and display a public friendship setting, but their value of the ”friends” entry is an empty list. This list is supposed to be populated with the befriended usernames pointing towards them.

The three preprocessing approaches provided in GraphTool_Preprocessing.py [3] present different ways of handling these anomalies. The first approach is the most conservative one in terms of data preservation. All 45.135 unidirectional ties are made bidirectional regardless of their cause. The second approach follows the first one and additionally removes all nodes for which holds true: not nominated as friend and friendship setting private. This way isolated nodes are removed from the network. The argument for this approach would be, that isolated nodes are not integrated in the community of the website and thereby assumingly significantly less active and relevant. Problems associated with this approach are addressed in section 6.1 Future Work & Limitations. The last approach follows approach two and additionally excludes the 187 faulty nodes and their 331 unidirectional FRIENDS_WITH edges. Problems associated with this approach are addressed in section Limitations 6.1 Future Work & Limitations, as well.

Chapter 5

Analysis

The social context of these assortativity analyses allows for an interchangeably usage of the terms assortativity and homophily in the following sections.

The results of the descriptive analysis and the assortativity analysis are based on the first introduced preprocessing approach in the previous section. This is reasoned by its conservative nature and the short comings of the other two approaches. GraphTool_Descriptives.py [3] is structured to enable a comfortable implementation of the other two preprocessing approaches as well.

Focusing on homophily between users based on their political believes, requires information about their friendships and their political stances. In the context of this database design, this requires the investigation of not only a subgraph of *User* nodes and their friendship relations, but also of the *Issues* nodes and their connection to the *User* nodes. Following this requirement a network of two node types and two edge types was extracted from the database and preprocessed. This lead to the inclusion of *User* and *Issues* nodes and FRIENDS_WITH and GIVES_ISSUES edges.

The majority of the following analysis is focused on the structure of the *User* nodes and their friendship edges. Once the *Issues* node features of interest are copied to the *User* nodes, they and the GIVES_ISSUES relation become meaningless in this analysis. This is done in a small additional preprocessing in the GraphTool_Assortativity.py file. Section 5.2 Assorative Mixing dives deeper into this topic.

5.1 Descriptive

The complete, preprocessed graph featuring both types of nodes and edges contains 90.696 nodes and 234.100 (unidirectional) edges. When differentiating between the to types of nodes, a network with 45.348 *User* nodes and 188.752 FRIENDS_WITH edges between them emerges. Concurrently a second network of 45.348 *Issues* nodes is identified. Both networks share the same number of nodes, since the database design was laid out to assigns one *Issues* node to each *User* node, containing the users political stances. Accordingly, there exist 45.348 GIVES_ISSUES edges connecting *User* nodes and *Issues* nodes. The former differentiated network is referred to as friendship network in the following

The complete friendship network consists of 28.737 components for which most represent isolated nodes, and thereby components with a size of one. Figure 5.1 displays the distribution more differentiated. Unfortunately, due to the logarithmic scale of both axis, two size categories and frequencies are not visible. They seem to be located to far on the x-axis to be represented accurately. Table 5.1 contains the complete data of all size categories and frequencies.

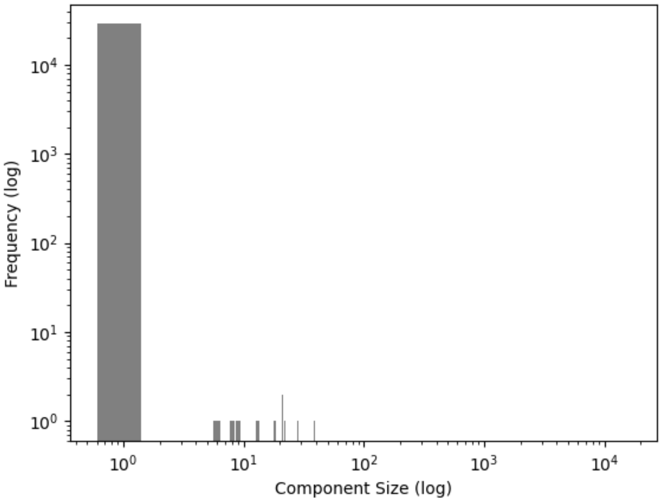

Figure 5.1: Component Size Distribution

Component Size	1	6	8	9	13	18	21	22	28	39	55	16382
Frequency	28726	1	1	1	1	1	2	1	1	1	1	1

Table 5.1: Component Size Distribution

The largest component of the friendship network contains 16.382 nodes and 187.478 unidirectional friendship edges. This accounts for 36,125% of nodes and 99,325% of the edges of the friendship network.

With 45.348 nodes and 188.752 edges, the friendship network shows a density of 0,00018%. Its larges component featuring 16.382 nodes and 187.478 edges displays a higher density of 0,0014%. Both appear to be rather sparse, in the context of online friendships. Possible explanations for this might be the focus on social interactions via debates. It is conceivable that the need for casual conversations is outsourced to other social media platforms. This would result in a small need for debate.org friendship interconnection. Another factor is assumably the optional private friendship setting, that masks edges that might otherwise be visible.

Figure 5.2 visualizes the degree distribution of the friendship network, while Figure 5.3 visualizes the degree distribution of the largest component. The reciprocal, unilateral nature of bilateral edges in GraphTool is accounted for. The analyzed degree is the indegree, which in this context is equal to the outdegree and equal to the number of bilateral edges, and hence the number of friendships a user has. Both are characterized with a maximum degree of 2.025, meaning that both networks contain a user that is befriended with 2.025 other users. The average degree of the whole friendship network falls on 4,16, while the mode and median degree is valued 0. The average degree of the largest component is higher with a value of 11,44. Its mode degree resides at 1, while its median degree is 2.

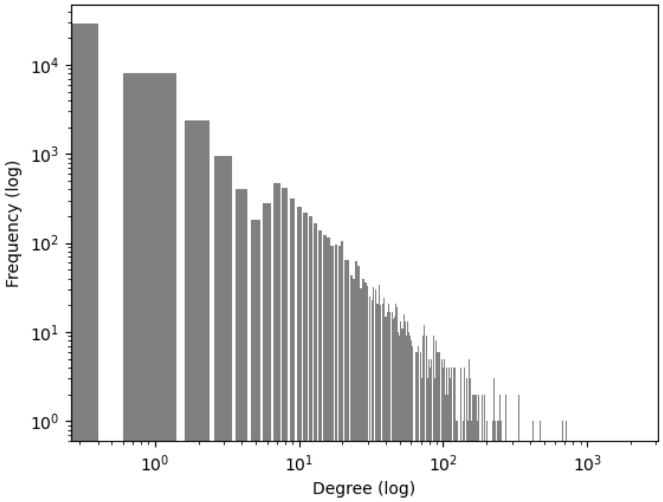

Figure 5.2: Degree Distribution

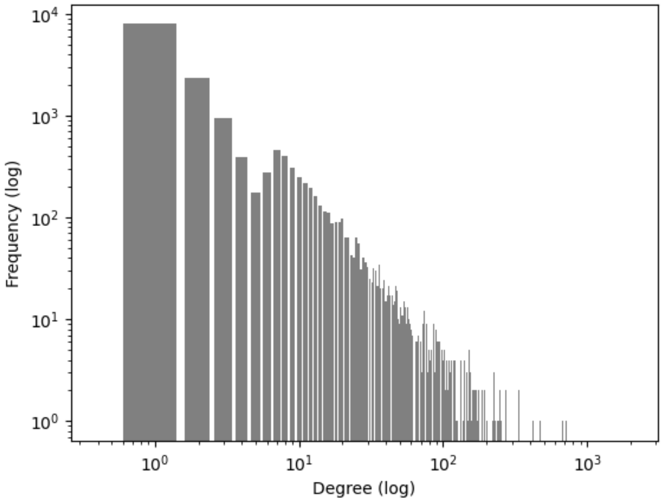

Figure 5.3: Degree Distribution - Largest Component

The friendship network displays a Global Clustering Coefficient of 0,1018 with a standard deviation of 0,0133 using the Jackknife method, while the largest components one, falls on 0,1016 with a standard deviation of 0,0133. These measures are so similar, since 99,325% of the friendship network edges are present in the largest component as well.

The diameter for the largest component is of length 14, while the calculation of the diameter for the whole network is impossible due to missing edges between components.

The value of the average Closeness Centrality of the largest component is computed as 0,277, while the median is 0,272 and the mode occurs 175 times with a value of 0,271. A more detailed view is provided in Figure 5.4. Figure 5.5 shows a Closeness Centrality visualization created with GraphTool.

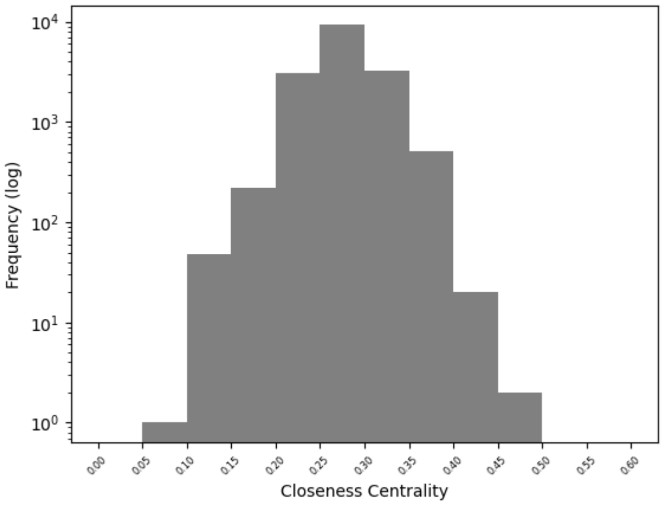

Figure 5.4: Closeness Centrality Distribution - Largest Component

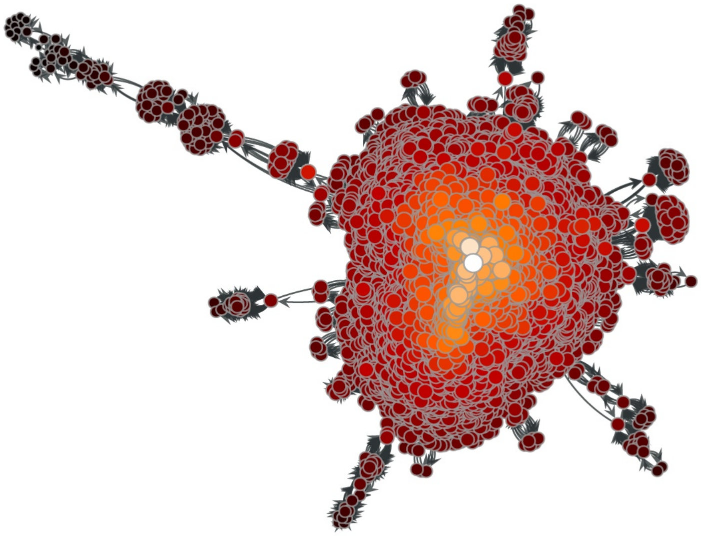

Figure 5.5: Closeness Centrality Visualization - Largest Component

In terms of Node Betweenness Centrality of the largest component, the average value falls on 0,00016, while median and mode are 0. The latter with a frequency of 9260. The distribution of Node Betweenness Centrality is presented in Figure 5.6, while the GraphTool visualization is found in Figure 5.7.

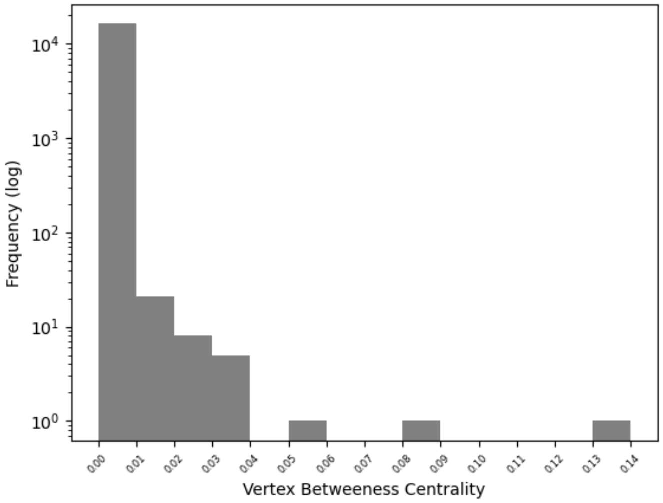

Figure 5.6: Node Betweenness Centrality Distribution - Largest Component

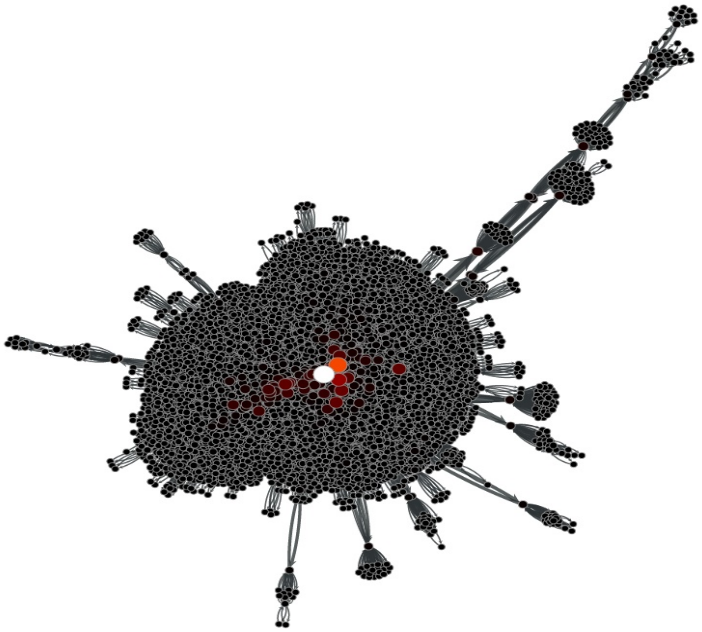

Figure 5.7: Node Betweenness Centrality Visualization - Largest Component

The last centrality measure and descriptive unit reported is the Eigenvector Centrality. The Eigenvector of the largest component is 3028,057. The average of the centrality falls on 0,0024, while median and mode are computed as 0,00027 and 1,19e-17. The latter with a frequency of 1. The actual frequency of nodes with a value close to 0 is believed to be higher. Figure 5.8 grants a more detailed view of the distribution, while Figure 5.9 visualizes the Eigenvector Centrality.

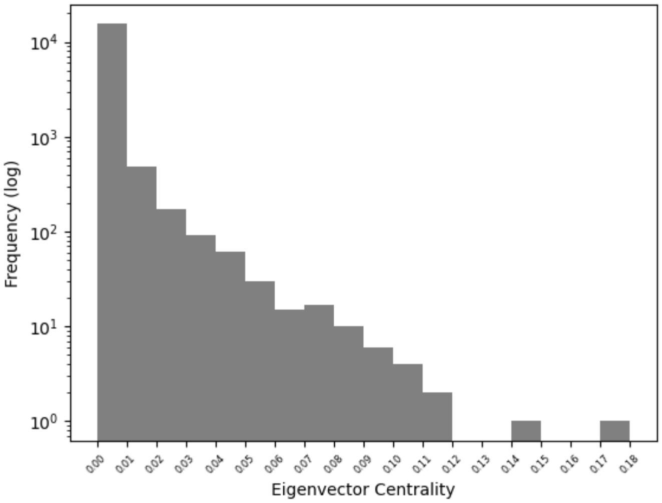

Figure 5.8: Eigenvector Centrality Distribution - Largest Component

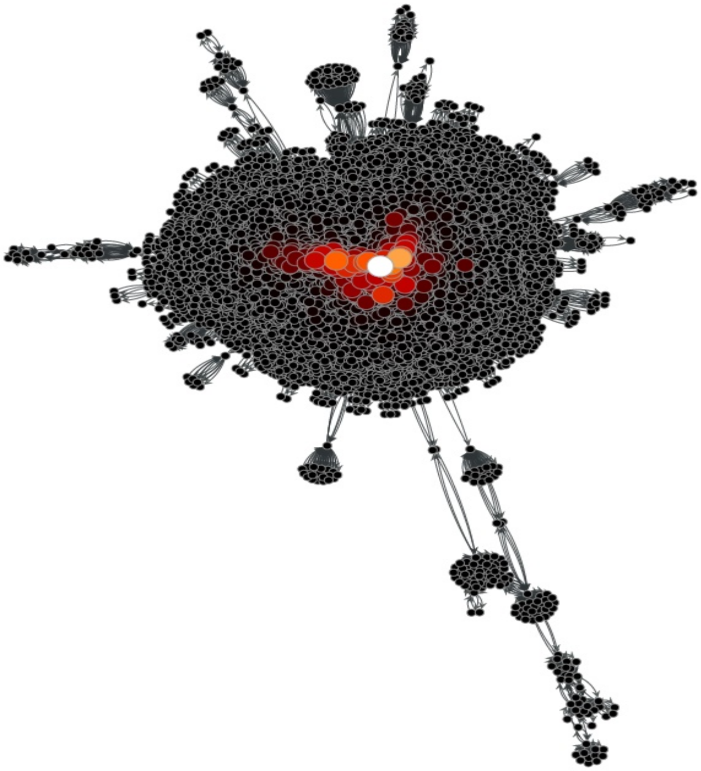

Figure 5.9: Eigenvector Centrality Visualization - Largest Component

5.2 Assortative Mixing

An investigating of Assortative Mixing in the friendship network, requires some additional preprocessing. As mentioned in section 3.1.1 General Concept, the database design saves the political stances of each user separate from the *User* node in their respective *Issues* nodes. Saving this information in the *User* nodes, instead of designing *Issues* nodes would ease the implementation of assortativity analyses such as the here presented. The reason why this design was not adapted is discussed in section 3.1.1 General Concept.

As first step of this smaller preprocessing, all stance relevant information from the *Issues* nodes is copyed to the *User* nodes. Subsequently, the *User* nodes gain additional node features containing the stance information. This information is obtained as unedited copy of the original string of the *Issues* node and as coded integer values. The main policies of interest for this analysis are the issues of abortion, gay marriage, the believe in (the existance of) global warming, and national health care. These issues were singled out because they are believed to be polarised enough to potencially inhibit interesting homophily patterns. On each of these stances users are able to answer with one of the following positions: Pro (in favor), Con (against), Und (undecided), N/S (not saying), N/O (no opinion). The two integer node features were coded to:
1 (Pro), 0 (Und),-1 (Con), -99(N/S, N/O) and
2 (Pro), 1 (Con), 0 (N/S, N/O, Und).

Reasons for these two coding approaches are on the one hand, that functions provided by the GraphTool library concerned with assortativity analysis do not handle negative scalar values well. On the other hand, the homophily of debate.org users is analyzed by differentiating between users of all stances, users of ”Pro”, ”Con” and ”Und” stances, and user of only ”Pro” and ”Con” stances.

Additionally a progressiveness score for each user is calculated. The idea is to give each user a score representing their overall progressiveness concerning the five focal topics. Starting at 0, each users ”Pro” stance adds 1 to their score, while each users ”Con” subtracts 1 from their score. This lead to values ranging from -5 (very conservative) to +5 (very progressive). Due to the mentioned constrain of GraphTool, a modified version of the score was established as well. It was moved by 5 points in the positive direction, resulting in a value range of 0 (very conservative) to 10 (very progressive).

For a calculation of the plain assortativity score, the issues socialism and the political ideology of users are considered as well. Socialism falls into the same value spectrum of the focal five Issues, while the political ideology range encapsulates ”Anarchist”, ”Apathetic”, ”Communist”, ”Conservative”, ”Green”, ”Labor”, ”Liberal”, ”Libertarian”, ”Moderate”, ”Not Saying”, ”Other”, ”Progressive”, ”Socialist” and ”Undecided”.

Table 5.2 presents the calculated Assortative Mixing scores. Almost all assortativity scores increase when narrowing down the scope of answers considered for the analysis. This is little surprising, since the opportunity for users to display friendships with users of other stances shrinks with the scope of considered answers. Naturally, the extreme of considering only users of one particular stance would result in an assortativity of 1. This is because in this edge case (hah!) only edges between users of the respective stance are able to be analyzed.

One exception to the steady increase is the decrease of Socialism assortativity when jumping from all values to ”Pro”, ”Con” and ”Und” values. This might be explained by a lot of users having ”N/S” or ”N/O” stances being befriended with users of the same stance. These user would positively effect the assortativity score when being considered in ”All” but not when not being considered in ”Pro,Con,Und”. Another interesting observation is the steadily low assorativit score for the issue of drug legalization. This might be explained by a lower underlying polarisation of users.

The limitations of this plain assortativity score are, that underlying patterns are not revealed. It is, for instance conceivable that the fairly high assortativity of the topic gay marriage is caused by equal tendencies of ”Pro” and ”Con” users to befriend users of the same opinion. However, it is also conceivable, that the high assortativity is caused by highly skewed tendencies. A ”really high” tendency of ”Pro” stance users to befriend other ”Pro” stance users could mask a fairly low tendency for ”Con” users to befriend other ”Con” stance users, or vice versa.

When extended to cases where more then two stances are considered, this ”really high” tendency might even mask fairly ”negative” tendencies of users for befriending users of different stances. In order to gain more insights into the homophily between users beyond its plain assortativity scores and to reveal potencially interesting patterns beneath, additional visualization is produced in the following.

With focusing on the issue of abortion, a subgraph of the friendship network with nodes with only strong opinions regarding the topic, is analyzed in more detail. Strong opinions are regarded as ”Pro” or ”Con” stances. This partition results in 6.877 nodes with a ”Pro” stance, 6.798 nodes with a ”Con” stance and 84.130 friendship edges between them. Figure 5.10 showcases how friendship edges are distributed between *User* nodes in this abortion subgraph of users with strong opinions. The first measure per quadrant reports the fraction of edges falling into a tile, relative to the overall number of edges. The second measure reports the abso-

	Coefficient (All)	Variance (All)	Coefficient (Pro,Con,Und)	Variance (Pro,Con,Und)	Coefficient (Pro,Con)	Variance (Pro,Con)
Abortion	0,062	0,00147	0,096	0,00277	0,120	0,00342
Gay Marriage	0,063	0,00151	0,112	0,00313	0,130	0,00363
Global Warming Exists	0,049	0,00147	0,069	0,00297	0,091	0,00412
Drug Legalization	0,038	0,00143	0,040	0,00268	0,057	0,00364
National Health Care	0,048	0,00143	0,076	0,00319	0,100	0,00410
Socialism	0,078	0,00146	0,070	0,00365	0,102	0,00533

Table 5.2: Assorative Mixing Coefficients and Variance

lute number of edges falling into a tile. As mentioned previously in this report, GraphTool handles the bidirectional nature of debate.org friendship relations as two reciprocal unidirectional edges. The actual number of friendships between Users with a conservative stance is thereby 23.654/2 = 11.827 and the number between progressive users is 23.464/2 = 11.732. The numbers of the other ”Pro-Con” and ”Con-Pro” tiles are identical. Both tiles represent one direction of the bilateral friendships between ”Pro” and ”Con” stance users in the network. Thereby 18.506 inter-stance friendships are present in the graph.

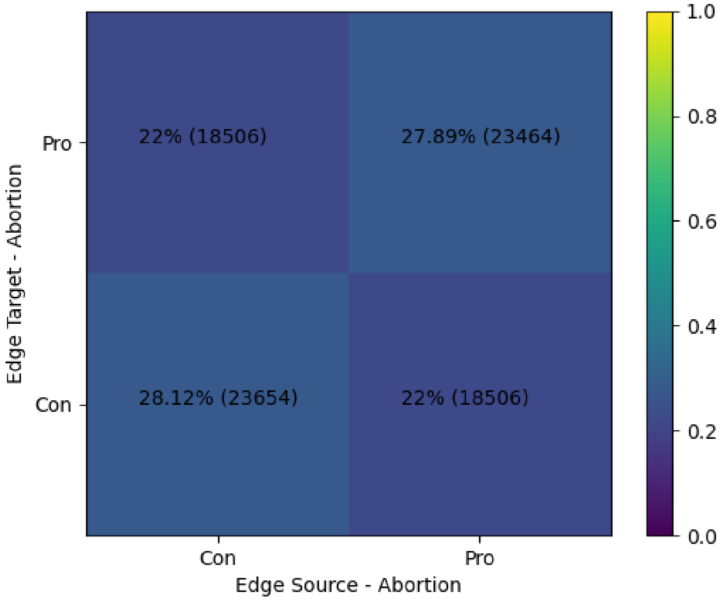

Figure 5.10: Friendship Edge Distribution - Abortion

Unfortunately this visualization focuses solely on the edge distribution and does not take the node distribution into account. It grants a more differentiated insight than the plain assortativity score, but still is far from optimal. Without a normalization of the edge numbers, cases with underlying skewed homophilic tendencies might not be reveal. A hypothetical case with 10.000 edges between progressive and conservative users and 10.000 between users of different stances is used to demonstrate this. A case with this even edge distribution but with e.g. 10.000 abortion conservative users and 100.000 abortion progressive users would be visualized as balanced, but conservative users would actually exhibit far grater tendencies for homophily then progressive users. The more user nodes a stance has, the more opportunities there are for same stance edges to emerge. Edges seizing their opportunities differently among stances, implicates differences in homophily.

Concerning this caveat, more information is provided. A best case scenario would include normalization for every tile, but due to the limited scope of this project, only the tiles concerning the edges between *User* nodes of the same stances are normalized. This normalization is done by the respective number of *User* nodes per stance. This approach is limited, but grants more context for the visualization nonetheless.

The normalized measure for edges between abortion conservative users is 23.654 edges / 6.798 nodes = 3,47955, while the one between abortion progressive users is 23.464 / 6.877 = 3,41195.

In cases where the visualization or the normalized measure is imbalanced, but the respective other is not, different, hidden homophilic patterns are expected and further investigation is necessary. This is true for extreme cases as well, in which both units are imbalanced, but in an opposite direction. In the case of abortion, both are fairly symmetric, implying no highly asymmetric, hidden homophily patterns.

The subgraph concerned with the issue of gay marriage includes 8.756 *User* nodes with a progressive stance, 3.618 *User* nodes with a conservative stance and 81.542 edges between them. The edge distribution in Figure 5.11 displays a asymmetry with 53.1% of edges emerging from friendships between progressive users. When considering the asymmetry of the normalized measures, this imbalance is likely resulted by the higher number of progressive users and the thereby higher number of friendship opportunities. The normalized measures are 9.044 / 3618 = 2,5 (Con-Con) and 43.298 / 8.756 = 4,945 (Pro-Pro).

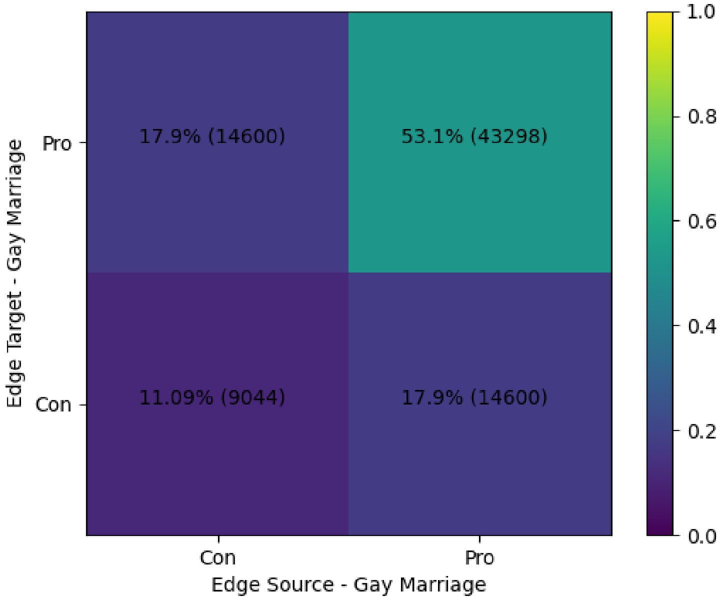

Figure 5.11: Friendship Edge Distribution - Gay Marraige

A network partition of only users with strong stances towards the believe in the existence of global warming consists of 2.804 users displaying a conservative stance on this matter, and 7.925 users displaying a progressive stance. 65.486 unidirectional friendship edges span between them. The visualization in Figure 5.12 shows a even more imbalanced distribution, with 59.88% of the friendship edges falling between progressive users. This asymmetry is once again found in the normalized measures as well with 4.750 / 2.804 = 1,694 (Con-Con) and 39.210 / 7.925 = 4,948 (Pro-Pro), indicating no hidden asymmetric homophily patterns.

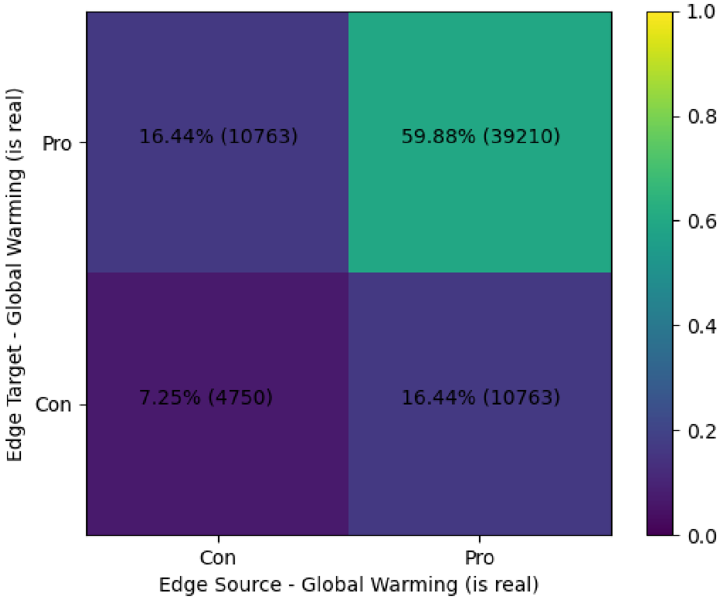

Figure 5.12: Friendship Edge Distribution - Believe in Global Warming

Figure 5.13 represents the visualization of edge distribution regarding the issue of drug legalization. The respective subgraph features 5.802 legalization conservative users, 6.207 legalization progressive users and 76.244 edges between them. The edge distribution is significantly more skewed then the distribution of the abortion context as well, but less skewed then the distributions concerning the previous two investigated topics. 39.64% of all all friendship edges fall into the quadrant of progressive users. The normalized measures of 12.100 / 5.802 = 2,086 (Con-Con) and 30.226 / 6.207 = 4,89 (Pro-Pro) indicate once again, that this asymmetry is mostly caused by the uneven distribution of opportunities, rather then by underlying and hidden homophily patterns.

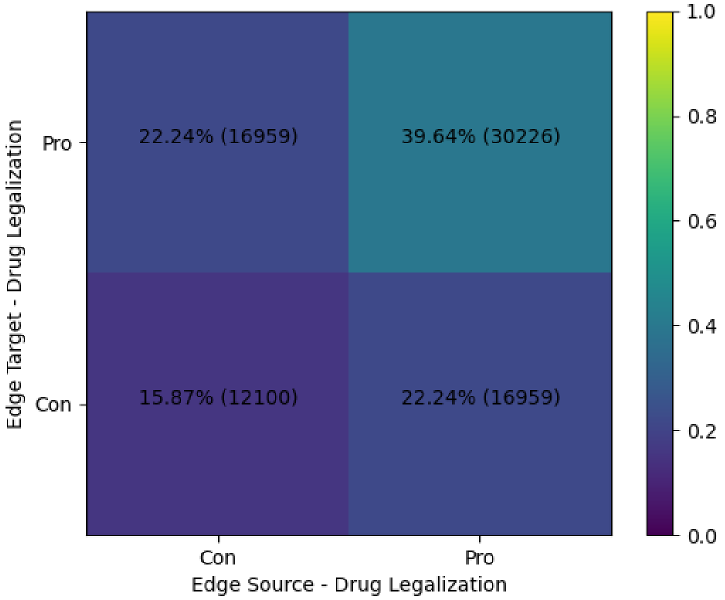

Figure 5.13: Friendship Edge Distribution - Drug Legalization

As last focal issue, the network partition of users with strong stances toward national health care consist of 3.206 opposing users, 5.873 supporting users and 59.208 total edges between them. Figure 5.14 shows, that the distribution is moderately skewed as well. With 35,34% of the edges falling into the tile of progressive users, a little less skewed than the distribution concerning drug legalization. Here the normalized measure of 12.218 / 3.206 = 3,811 (Con-Con) and 20.926 / 5.873 = 3,563 (Pro-Pro) reveal a more interesting pattern. Not only is the ration not skewed toward ”Pro-Pro”, in a similar matter as observed in the visualization, but it is skewed into the opposite direction. This is a first indication for higher homophilic tendencies between national health care conservative users than national health care progressive users.

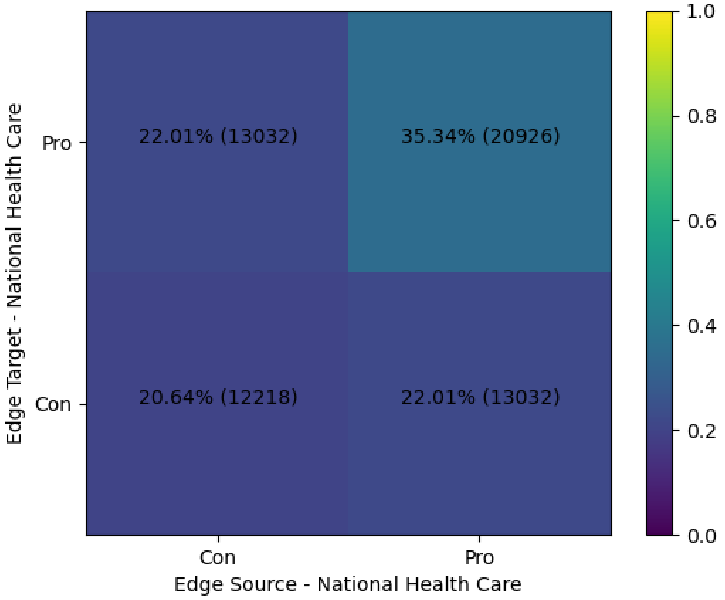

Figure 5.14: Friendship Edge Distribution - National Health Care

The calculated progressiveness score using these 5 focal topics (following PS), was analyzed similarly. A corresponding network partition including all users that exhibit at least one strong stance (meaning ”Pro” or ”Con”) on one of these five issues, is generated. The PS distribution of users of this partition is displayed in Figure 5.15. 118.534 friendship edges emerged between all users of this partition. Figure 5.16 visualizes the friendship edge distribution between users of the various score categories. The small size of the tiles, made the labeling of relative and absolute edge numbers unfeasible. Nonetheless a general skewness towards more progressive users is observable once again. A non-intuitive and for now puzzling pattern visible, lies in the gradually increasing fraction of edges along the PS. The increase differs for even and odd number of the PS. Figure 5.17 shows the distribution of the normalized measures. The normalized measures reflect the pattern of of more edges emerging between user of even Progressiveness Scores, but do not explain the asymmetry towards PS 10 completely. Here not only a deeper analysis of varying homophilic tendencies deems necessary, but also a investigation into the usefulness of the PS.

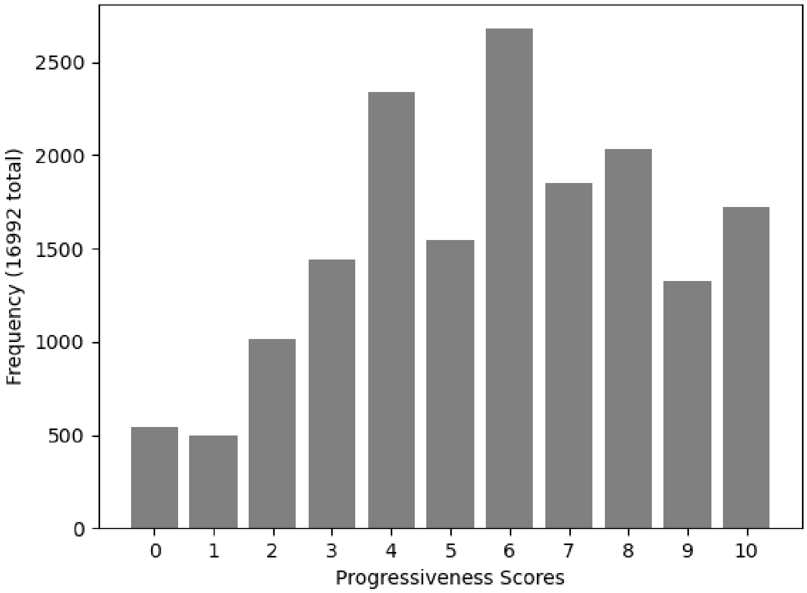

Figure 5.15: Progressiveness Score Distribution

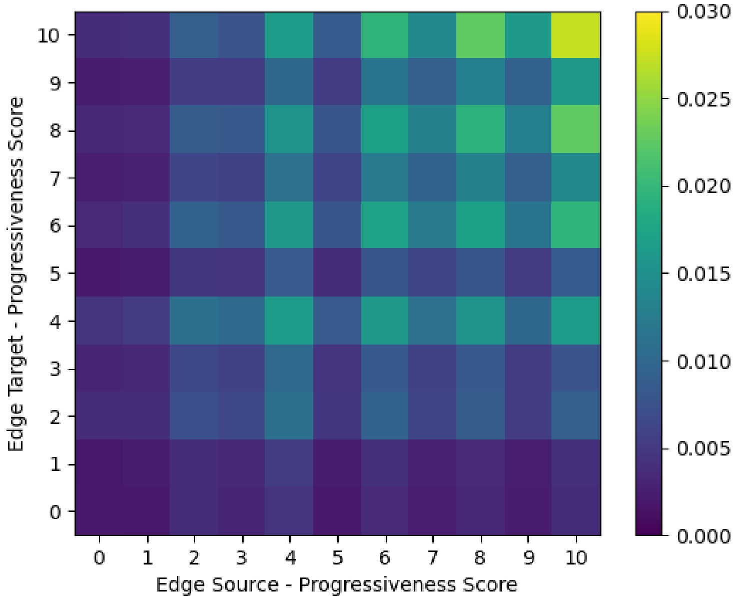

Figure 5.16: Friendship Edge Distribution - Progressiveness Score

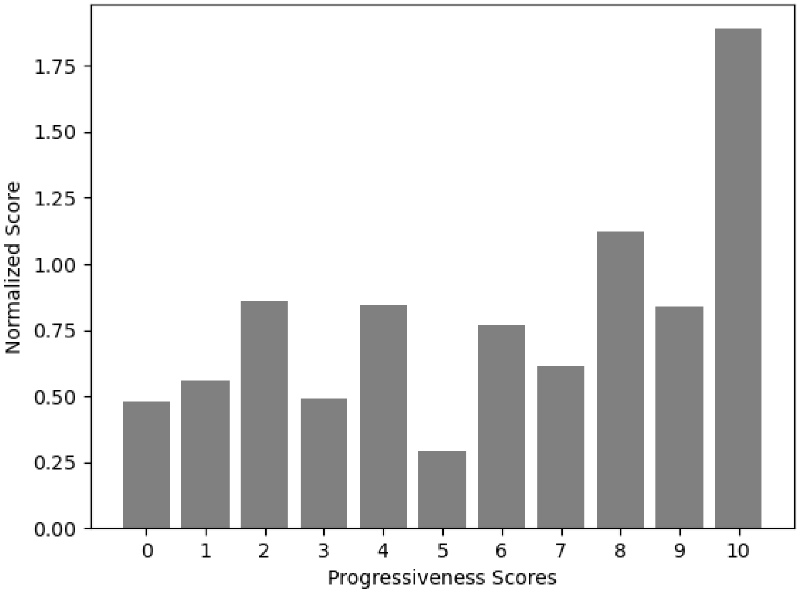

Figure 5.17: Progressiveness Score Normalization Distribution

Chapter 6

Conclusion

During this project a Neo4j graph database was successfully conceptualized, created and populated with data from the online debating platform debate.org. Subsequently, a subgraph of user friendships and political stances was extracted and analysed using the python library GraphTool. Five political issues investigated closely were abortion, gay marriage, the believe in global warming, drug legalization and national health care. On all five dimensions homophily was observed to a small extend. The analysis of the issue of national health care indicates homophily patterns that requires additional, more sophisticated measures, than the ones used in this report. Additionally a progressiveness score was build, combining stances of the five issues to one measure. This new formula is supposed to illustrate a general progressive or conservative stance of users. Following the approach used for the five focal issues, the assortativity of this score was analyzed as well. It shows small homophily and indicates a benefit of deeper analytical dives.

During the course of this project I was able to familiarize myself with Neo4j and GraphTool and developed an intuitive understanding regarding both. Additionally I gathered experience in designing graph databases, employing the query language Cypher and in utilizing Docker as a way of overcoming compatibility issues regarding Python libraries. Last but not least I was able to deepen my knowledge in Python and practice network analysis in this context.

6.1 Future Work & Limitation

In pursuing this project further, several processes could be optimized. One optimization target might be the population of the Neo4j graph database. With the employment of additional functions of the APOC Neo4j library this population could be sped up significantly.

In terms of data accuracy, the authors of the data set might be contacted. This way the limitation of possibly inaccurate inferred relative time data, mentioned in section 3.1.2 Time Dimension could be addressed. The retrieval of an accurate day or time window of the data scraping process would result in a more accurate representation of the time dimension in the graph database. Regardless of this, the implemented *Timeline* nodes of the graph database could be adjusted to a finer grain. Currently they are grained on a year basis. As last but most striking point, the comment time dimension might need to be handled differently, when investigating issues concerning it. Currently, the only relative-given time dimension of comments (e.g. ”6 months ago”) is handled by inferring a date of comment creation. This is done by subtracting the relative-given time of 2017. The year 2017 marks the latest date of the scrapped time window (on year grained basis). The fact, that there exist debates with non-relative creation date 2018, renders some classification of comment creation faulty by at least one year.

Focusing on the preprocessing, further approaches might be conceptualized, tested and implemented in order to find a more appropriate way of handling the asymmetric friendship relations in the original data set. The current exclusion of only isolated nodes with a private friendship setting seems arbitrary. In cases where such an exclusion deems necessary, excluding all isolated *User* nodes independent of their privacy settings, seems more appropriate.

In terms of a descriptive analysis of the data set, more centrality and network measures in general might be included, if beneficial.

For the analysis of assortativity, adding additional political issues might reveal more interesting patterns of homophily. Developing a more sophisticated way of edge normalization between nodes of different stances might give away more interesting homophily tendencies not observed with the presented approach. An investigation into the nature of the progressiveness score might verify its usefulness or raises new measures of greater accuracy.

Shifting the focus of analysis to data less related to assortativity and friendships could be of great interested as well. These newly focused analyses could incorporated the semantics of debates, opinions and polls.

At first glance, there seem to be a lot more users with progressive stances than conservative users, see Figure 5.15. However, the small sample of occasionally, by the author observed polls, opinions and sometimes debate title imply very conservative world views. It could be worthwhile to investigate to which extend these entities are created by trolls, to which extend the user base is left or right leaning and finally to analyze to which extend conservative voices might be louder then progressive and moderate ones in the context of debate.org and online discourse in general.

Another point of investigation might be the sample size of the data set mentioned already in section 2.2 Data Set Features . Depending on, whether the 6,55% of overall debate.org users featured in the provided data sets, really do represent the most active ones, interpretations of future results might be adjusted.

On a smaller scale of limitations and future work, a more consistent naming scheme in the provided code files and in the general database design might be pursued. Exemplary proposals would be the renaming of the FRIENDS_WITH relation to GIVES_FRIENDSHIP, or the DEBATES_IN relation to GIVES_DEBATE.

Bibliography

[1] Debate.org. https://www.debate.org/. Accessed: 2021-02-02.

[2] Debate.org demographics. https://www.debate.org/about/demographics/ . Accessed: 2021-02-02.

[3] Github alexander haberling
neo4j_debates.org. https://github.com/Ahaberling/Neo4j_Debates.org. Accessed: 2021-02-02.

[4] Graphtool. https://graph–tool.skewed.de. Accessed: 2021-02-04.

[5] Neo4j graph database. https://neo4j.com. Accessed: 2021-01-28.

[6] Esin Durmus and Claire Cardie. A corpus for modeling user and language effects in argumentation on online debating. *arXiv preprint arXiv:1906.11310*, 2019.

[7] Esin Durmus and Claire Cardie. Exploring the role of prior beliefs for argument persuasion. *arXiv preprint arXiv:1906.11301*, 2019.

Appendix A

List of political issues

The 48 political and moral issues featured on the website and in the data sets:

Abortion, Affirmative Action, Animal Rights, Barack Obama, Border Fence, Capitalism, Civil Unions, Death Penalty, Drug Legalization, Electoral College, Environmental Protection, Estate Tax, European Union, Euthanasia, Federal Reserve, Flat Tax, Free Trade, Gay Marriage, Global Warming Exists, Globalization, Gold Standard, Gun Rights, Homeschooling, Internet Censorship, Iran-Iraq War, Labor Union, Legalized Prostitution, Medicaid Medicare, Medical Marijuana, Military Intervention, Minimum Wage, National Health Care, National Retail Sales Tax, Occupy Movement, Progressive Tax, Racial Profiling, Redistribution, Smoking Ban, Social Programs, Social Security, Socialism, Stimulus Spending, Term Limits, Torture, United Nations, War in Afghanistan, War on Terror, Welfare

Appendix B

List of votemap items

List of items in each votemap:

- Agreed with before the debate
- Agreed with after the debate
- Who had better conduct
- Had better spelling and grammar
- Made more convincing arguments
- Used the most reliable sources
- Total points awarded

Appendix C

Lines for GraphTool

Manually added lines for GraphTool implementation of Neo4j Graphml files:

```
     <key id=”labels” for=”node” attr.name=”labels” attr.type=”string”/>
<key id=”label” for=”edge” attr.name=”label” attr.type=”string”/>
```

Ehrenwörtliche Erklärung

Ich versichere, dass ich das beiliegende Individual Project ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Ich bin mir bewusst, dass eine falsche Er- klärung rechtliche Folgen haben wird.

Mannheim, den 05.02.2021

Unterschrift

A. Habeling