Understanding Twitter Network of the 2016 Presidential Election

Min Hyung (Daniel) Kang

March 8th, 2015

Abstract

In this paper we consider the tweets posted by Twitter users on main candidates for the 2016 U.S. Presidential Election, with the goal of extracting as much information as possible from the social media. Networks are constructed from scraped data, and various evaluations such as spectral clustering are performed to compare different networks. Non network-specific analyses, such as sentimental analysis and Word2Vec are also considered. We observe that the networks follow characteristics of a scale-free network, as well as that they accurately depict the relevant background information of the collected dataset in various manners.

1 Introduction

1.1 Online Social Network and Twitter

In the 21st century, notably within the past decade or so, social networking websites have become a major part of human interactions. A research conducted by Pew Research center indicates that the proportion of people using social networks increased from 7% in 2005 to 65% in 2015, and for young adults aged 18-29 this number skyrocketed to almost 90% [1]. An article by Domo, a data access management company, states that the users now produce and share a surprising amount of content: 2,460,000 pieces of Facebook content, 277,000 Twitter posts, and 216,000 Instagram photos every minute [2]. It is easy to imagine that so much data may be able to tell interesting stories, yet it is not a trivial task to extract such stories. Twitter, a relatively simple SNS application that allows users only basic functionalities (tweet, retweet, reply), is an example of such mass-producer of information.

The shares of Twitter may have decreased by 55 percent in 2015 [3], but the number of the platform's monthly active users has still passed 307 million back in the 3rd quarter of 2015 [4]. Every second, this website generates a massive dataset that can be utilized to serve various purposes. Hence, many companies are using Twitter to establish advertisement strategies, enhance user experience, and cut operational costs. On its website, Twitter introduces case studies of companies including Hilton, CBS News, and NBA, explaining how each industry took these 140-word posts to enhance its services and products [5]. As such, the data on Twitter is far more than an arbitrary collection of people's musings: the users and tweets form networks of a great complexity that, once studied, have a great potential of providing new insights to social interactions. Though individual tweets may be trivial, the constructed network may reveal a bigger picture that shares with us an insight about everyday life.

1.2 Past Analyses on Twitter

Given its massive dataset and relative ease of access with a well defined API(Application Program Interface), it is not surprising that numerous studies have been already done on Twitter data. For example, a research done at Arizona State University developed a diffusive logic equation for modelling the information diffusion on Twitter and obtained 97.64% accuracy on the test data [6]. Back in 2010, researchers at Microsoft have tried to predict the probability of retweet using features such as the number of followers [7]. Other researchers have done comparisons between different platforms. For example, a study done at USC Viterbi compares the information cascade in Twitter and Digg, a news aggregator website, establishing models and providing various network statistics and analyses [8].

As such, Twitter has been studied from various directions across different fields and topics. However, a noteworthy aspect of Twitter is that each network created from different dataset is unique. Of course, the networks may be generalized as they were done at Pew Research Center back in 2014, when the researchers classified Twitter topic networks into 6 groups, from polarized crowd discussing political topics to broadcast network for breaking news [9]. Though such classification might be possible, a generalization might miss the unique story which each network tells. The story that we attempt to understand in this paper is that of the 2016 U.S. Presidential Election.

1.3 Political Campaign Background

The 58th U.S presidential election is scheduled for Tuesday, November 8, 2016. As of the time of writing of this paper (Feb-March 2016), seven candidates are still in the running for their respective parties. They include Hillary Clinton and Bernie Sanders from the Democrats, and Ben Carson, Ted Cruz, John Kasich, Marco Rubio, and Donald Trump from the Republicans [10]. This paper focuses on three of the mentioned candidates: Hillary Clinton, Bernie Sanders, and Donald Trump. The candidates were chosen according to the amount of observed public attention, the sample of two democrats and one republican becoming the unintended somewhat unbalanced result.

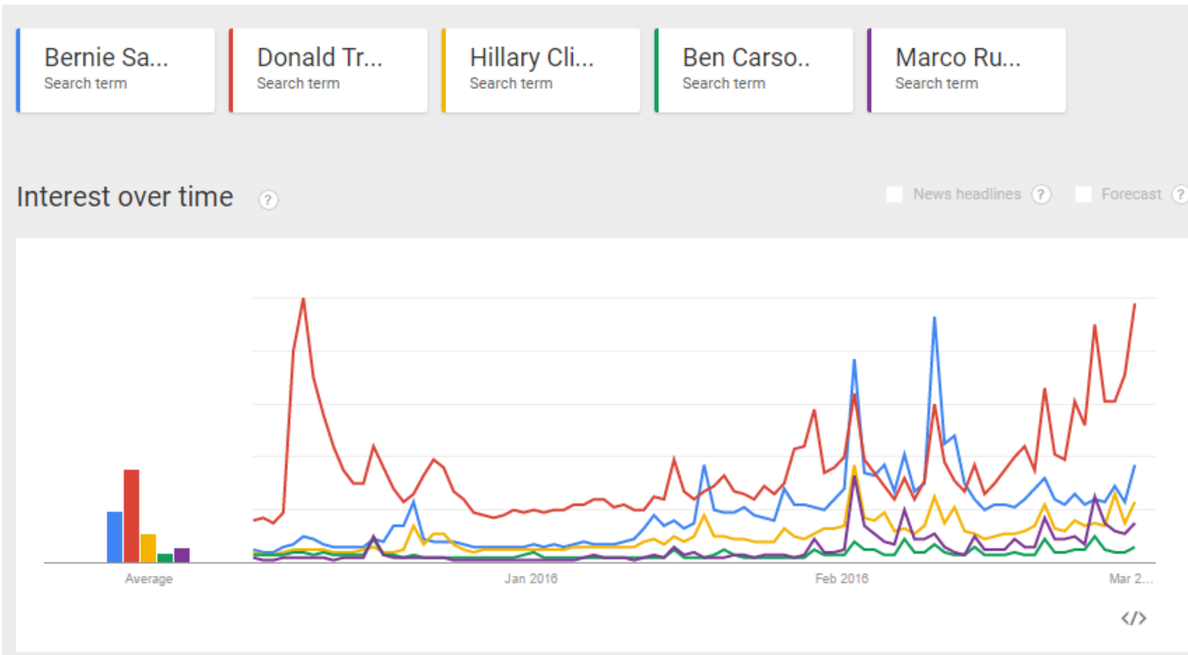

Figure 1: Google Trends showing search rates of 5 candidates between December 2015 - February 2015.

As shown in Figure 1, these three candidates are the most searched candidates of the 2016 Presidential Election. Accordingly, the number of followers for their social media accounts have greatly increased as well, portraying the sharp increase of people's interest on these candidates.

Platform	**Date**	**Candidates**		
		Hillary Clinton	Bernie Sanders	Donald Trump
Twitter	Sep 28th, 2015	4,340,000	598,000	4,310,000
	March 4th, 2016	5,289,500	1,304,660	5,947,030
	Change (%)	949,500 (22%)	706,660 (118%)	1,637,030 (38%)
Facebook	Sep 28th, 2015	1,427,180	1,600,866	3,868,917
	March 4th, 2016	2,684,440	3,051,020	6,138,000
	Change (%)	1,257,260 (88%)	1,450,154 (91%)	2,269,083 (59%)
Instagram	Sep 28th, 2015	317,000	231,000	435,000
	March 4th, 2016	902,000	941,000	1,100,000
	Change (%)	585,000 (185%)	710,000 (307%)	665,000 (153%)

Table 1: Increase of followers on social media for each candidate. Data for September 2015 is from intermarkets.net [11]. For Instagram and Twitter, the number of 'followers' was considered, while for Facebook the number of 'likes' was considered.

Hence, this paper focuses on the aforementioned three candidates and explores their individual networks. Note that this paper does not endorse a particular political party nor a specific candidate.

2 Data

We use python script and Twitter API to collect tweets that included the names of the political candidates in their text. As Twitter is a widely studied website, it provides an extensive API that developers or researchers can use to collect and study the data. Of course, there are some limitations to using the dataset:

- Twitter limits the number of query results per 15 min window, the rate being different for each type of request [12].
- Twitter sets a clear restriction on disclosure of private user information [13].
- Twitter only allows search of tweets published in past 7 days [14].

To meet these requirements, python script was run as long as possible, collecting 100 tweets per minute for each candidate. Data was collected from January 30th, 2016 until the script hit the limit and had to stop. In discussing the dataset, information that deemed to be private is left out or replaced with other text. Public figures and media accounts, however, are retained.

For two candidates, Hillary Clinton and Bernie Sanders, we were able to collect one week-worth of data. However, for Donald Trump, we were only able to collect tweets of four days worth before we hit the aforementioned time limit. The fact that the number of this four-day worth of tweets is still more than that of the tweets collected for a week for other candidates shows how frequently Donald Trump was tweeted about. The summary of the collected data can be found at Table 2.

The tweet frequency of each network shows a periodic distribution in Figure 2, reaching its minimum during the night. Across the time window, we can observe a few cases of outliers that stand out in a vertical line. They may be understood as an article that is quickly shared, or people talking about an event that just became available to the public.

Looking at the distribution of tweets including the original tweets of the retweets in Figure 3, we see that the dataset has few outliers, producing a graph with most tweets clustered to the right. Such outliers is examined more closely in section 3.3.

	Candidates		
	Hillary Clinton	**Bernie Sanders**	**Donald Trump**
Number of tweets	377,432	419,130	525,950
Number of Distinct Users	169,322	177,953	268,368
tweets / User	2.229	2.355	1.960
Number of Distinct Users* (including retweet/reply/mention)	175,372	186,040	280,582
Earliest tweet (UTC)	Mon Jan 25th, 2016 21:59:15	Mon Jan 25th, 2016 21:58:31	Thu Jan 28th, 2016 01:36:48
Latest tweet (UTC)	Sun Jan 31st, 2016 01:10:29	Sun Jan 31st, 2016 01:13:46	Sun Jan 30th, 2016 17:45:33
Earliest retweeted tweet** (UTC)	Tue March 18th, 2008 04:53:22	Wed Jan 26th, 2011 02:54:53	Mon May 4th, 2009 18:54:25
Duration (min)	7,391	7,395	3,849
Number of tweet / min	51.07	56.68	136.65

Table 2: Summary of collected data. *Includes all the users that tweeted as well as the users that were mentioned/retweeted/replie to. **The original post time of the earliest tweet that was retweeted during the collection period.

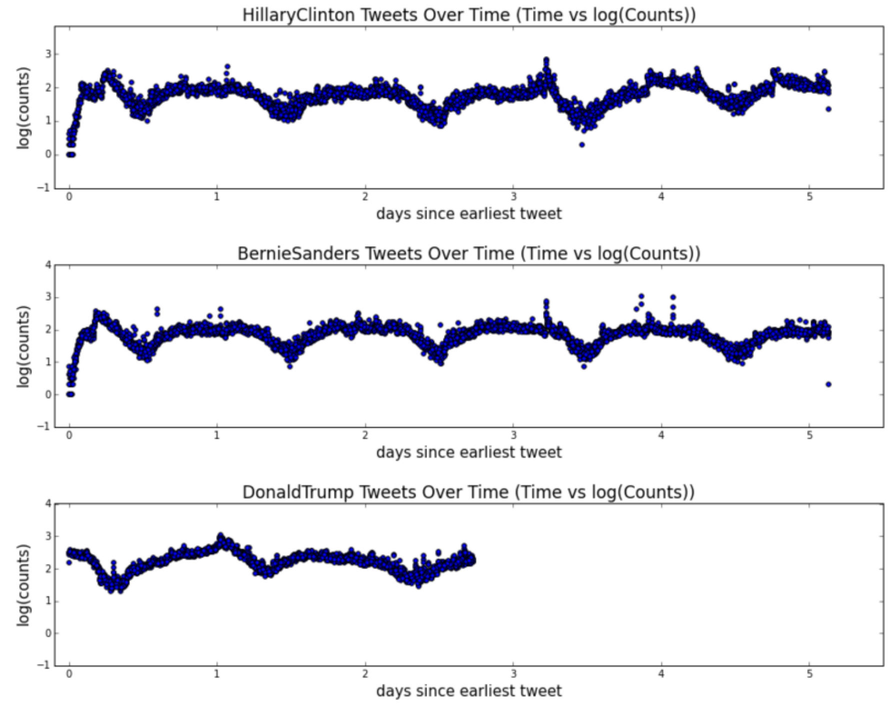

Figure 2: Distribution of tweets of each network, (days vs log(counts)). Day 0 for each candidate is respectively : Mon, January 25th 5pm (Clinton) / Mon, January 25th 5pm (Sanders) / Thu, January 28th 8:30pm (Sanders).

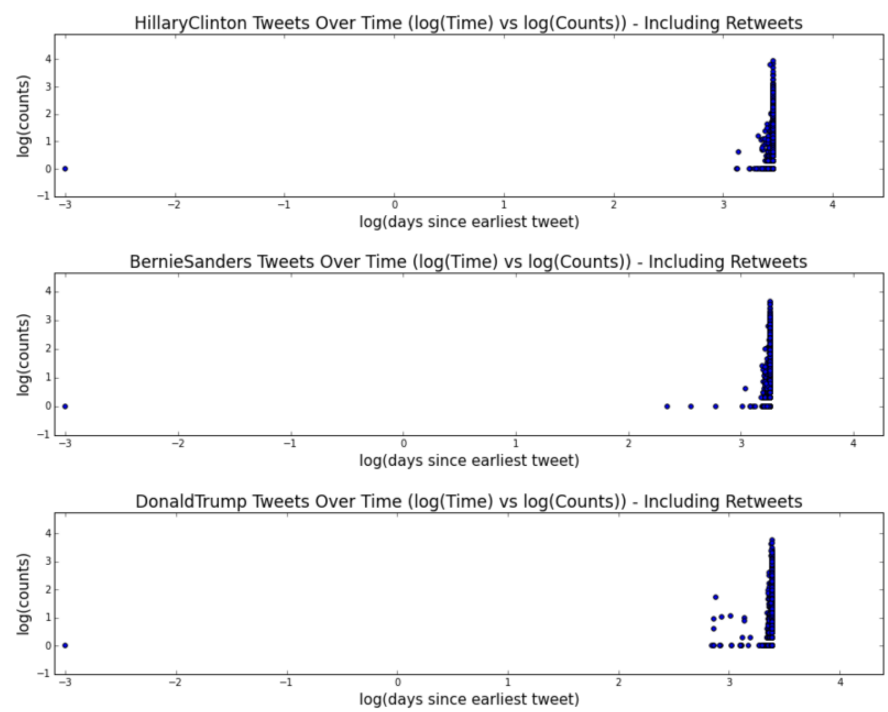

Figure 3: Distribution of tweets of each network across time including original tweets that were retweeted during data collection period, (log(days) vs log(counts)). Earliest tweet for each candidate is respectively : Tue, March 18th 2008, 12am (Clinton) / Tue, January 25th 2011, 10pm (Sanders) / Mon, May 4th 2009, 2pm. (Sanders)

3 Network Generation

There are three major networks that can be constructed from the collected data: mention, reply, and retweet networks. As one can easily see, each network corresponds to a different functionality of Twitter. It is natural to think that people may utilize each functionality differently, hence resulting in construction of distinct types of networks.

In each network, each node represents a unique user. We add a directed edge from user A to user B in the following cases :

- A retweets B
- A mentions B
- A replies to B

If user A retweets, mentions, or replies to user B more than once, we add up all the occurrences. It is noteworthy that if a researcher is interested in the spread of the information rather than the representation of the user activities of Twitter, one may reverse the direction of the above definition.

A sample construction of an example network is described in Figure 4. For each network, we only add the edges for corresponding functionality between the users. In the case of the combined network, we treat each functionality to have equal importance, and add up all the counts of user A retweeting, replying to, or mentioning B to determine the weight of edge from A to B.

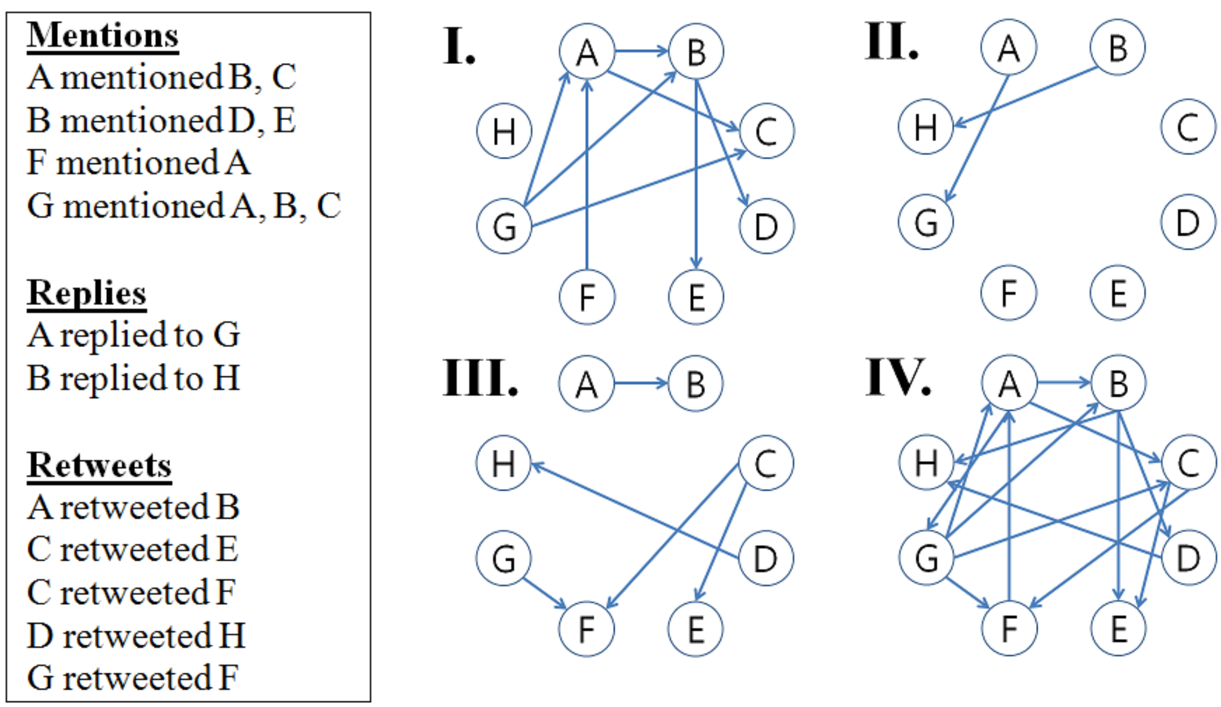

Figure 4: A toy example explaining the construction of each network, given the small dataset on the left. I. Mention Network II. Reply Network III. Retweet Network IV. Combined Network. Note that the weight of A→B edge will be 2 in the combined network as we add counts from mention and retweet.

The summary of constructed edges for each subnetwork (retweet, mention, reply) can be found in Table 3. It is worth noting that the number of reply edges is significantly smaller than the other two, and considering the fact that a user may mention many users but retweet only one, the proportion of retweets is very high.

	Candidates		
	Hillary Clinton	**Bernie Sanders**	**Donald Trump**
Number of Retweet Edges (%)	226,678 (60.06)	247,118 (59.00)	306,459 (58.27)
Number of Distinct Retweet Edges (%)	205,580 (54.47)	219,953 (52.48)	281,639 (53.55)
Number of Reply Edges (%)	12,550 (3.33)	15,030 (3.59)	21,998 (4.18)
Number of Distinct Reply Edges (%)	10,848 (2.87)	13,241 (3.16)	19,571 (3.72)
Number of Mention Edges (%)	342,769 (90.82)	401,092 (95.70)	468,707 (89.12)
Number of Distinct Mention Edges (%)	295,023 (78.17)	339,192 (80.93)	411,716 (78.28)
Number of Total Unique Edges (%)	295,764 (78.36)	339,930 (81.10)	413,056 (78.54)

Table 3: Summary of collected data. Percentage denotes the proportion of edges with respect to number of tweets.

3.1 Exploratory Analysis

A common feature of a complex network studied these days is that many of them are scale-free. That is, the degree distribution follows a power law:

$$p(k) \propto k^{-\gamma} \tag{1}$$

Hence, the first aspect of the network we check is whether this network may be considered to be scale-free.

As each network is constructed with directed edges, both in-degree and out-degree distributions are plotted.

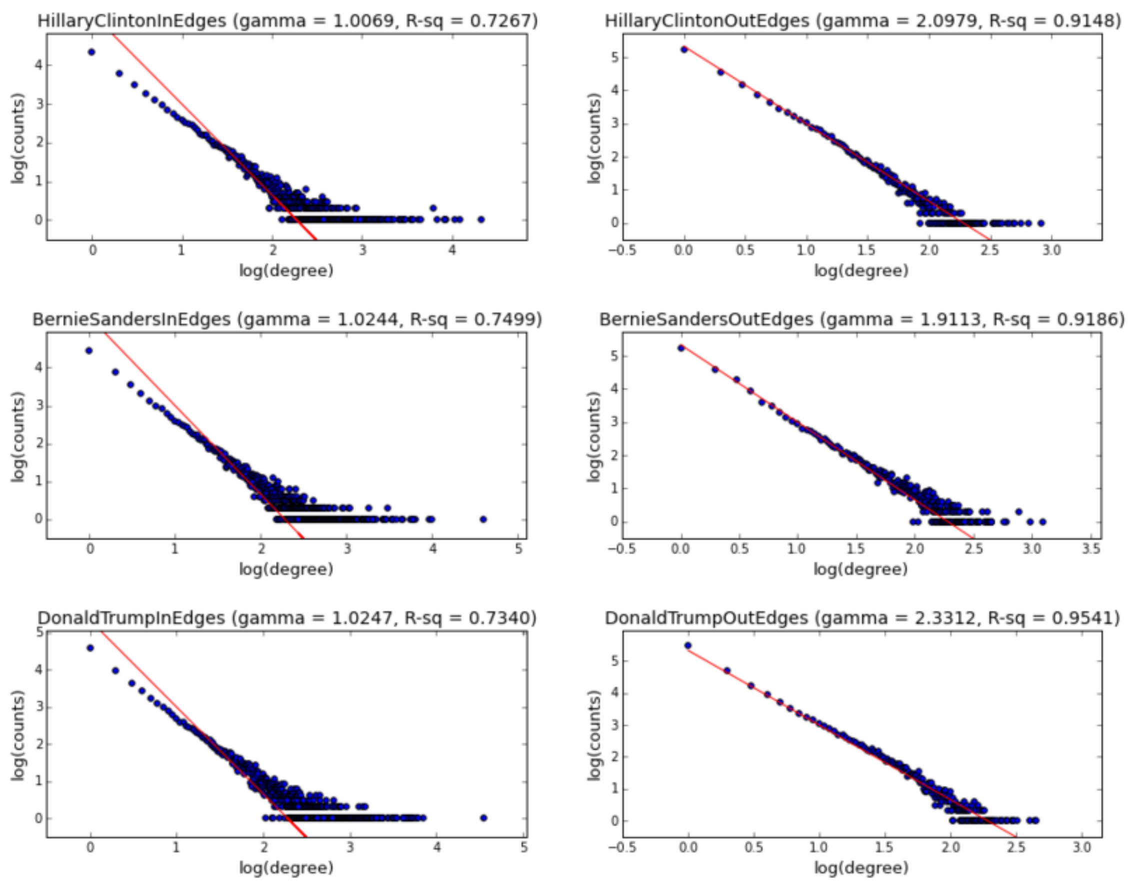

Figure 5: In/Out-degree distribution of each candidate (log-log scale). Linear regression is fit using python's default polyfit function.

Plotting them in a log-log plot, we observe a linear relationship with a fat tail, which is what we would expect from a scale-free distribution. A typical scale-free network observed have γ value between 2 and 3 [15]. One can observe that while the outdegree distributions have γ value similar to 2, indegree distribution portray much lower value, around 1. However, using a standard least-squares linear regression, it is easy to note that the fat tail behavior of the distribution greatly affects the slope of the line. From above plot one can visually confirm the discrepancy between the slope and the dataset. Hence, we introduce a different method to fit a power law distribution. Code by Clauset et al. utilizes maximum likelihood fitting methods with goodness-of-fit tests based on Kolmogorov-Smirnov statistic and likelihood ratios to fit a better power-law model to the empirical dataset.[16] The result is shown in Figure 6.

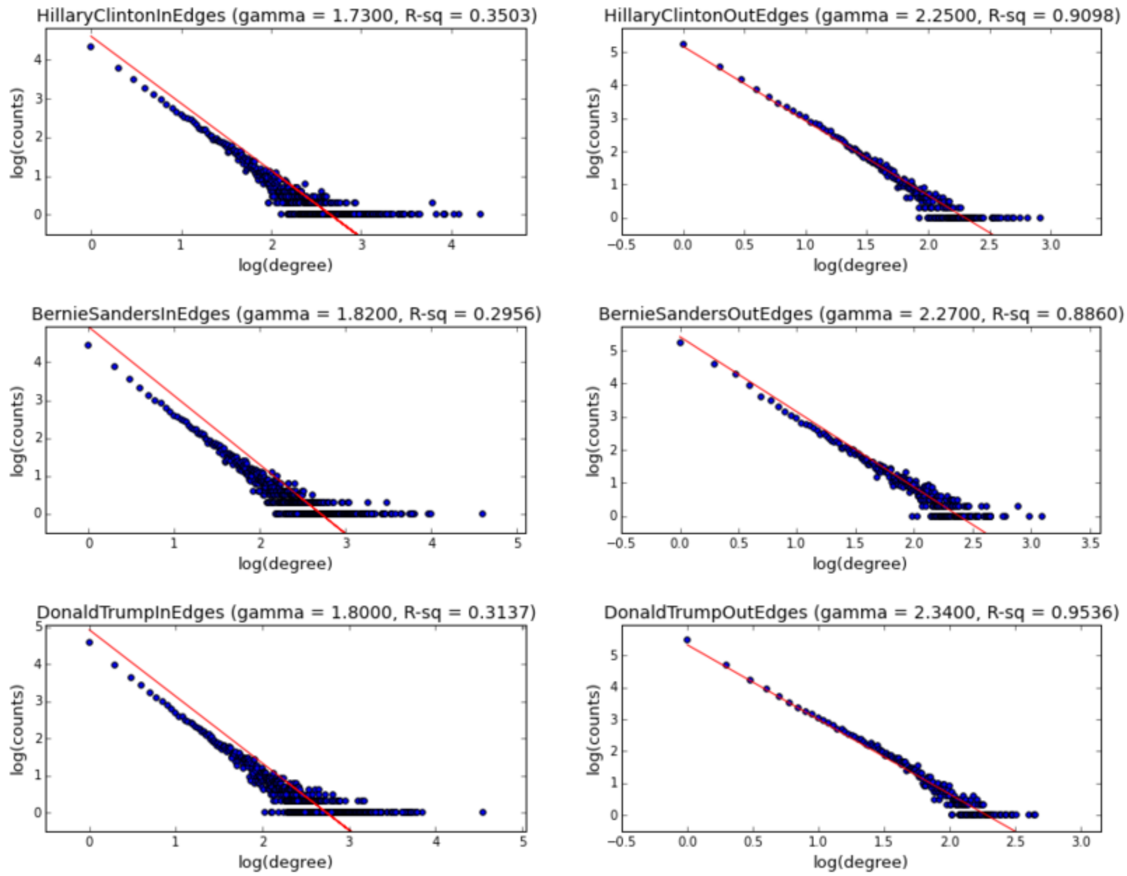

Figure 6: In/Out-degree distribution of each candidate (log-log scale). Linear regression is fit using Clauset et al.'s power-law distribution fitting algorithm [16].

One can see that the slope now fits the dataset much better. The slope of the out-degree distribution is between 2 and 3, while that of in-degree distribution is much closer to 2 than before. Computing the goodness-of-fit, however, we see that R-square values of this new method is lower than that of linear regression. Again, this is due to the fat tail behavior, which gives us large residuals. We note, therefore, that R-square is not necessarily a good measure of goodness of fit when it comes to fitting power-law models.

3.2 Mention Network

As one may mention many users in one tweet, and mentioning a user is a common behavior in Twitter, there is a great number of mention edges. User mentions can be understood as hashtags, and often contain information about the content of the tweet. It is not surprising, therefore, that each candidate is mentioned the most in their own networks. As one can see from appendix A.2, significant users mentioned are the candidates or major media such as CNN and The New York Times. This is reasonable, for when people talk about the candidates there is a high chance that they 'mention' them, and the context is usually the report or an article on media.

The only significant users that are often mentioned and not related to media nor candidates are Nathan Zed in Clinton network and Vampire Weekend in Sanders network. The reason they are mentioned can be easily discovered by looking at the events that occurred around the time of the tweets. On January 25th, Nathan Zed tweeted 'Hillary Clinton's latest message to the youth,' which was a satirical video of him disguised as Hillary Clinton. This video, as of February 23rd, was retweeted more than 14,500 times and received 22,000 likes. Hence, it is not surprising that tweets collected about Hillary Clinton between Jan 25th and Jan 31st include a lot of mentions of Nathan Zed. On the other hand, Vampire Weekend is a rock band that Bernie Sanders sang with in one of his rallies in Iowa City on January 30th [17]. As such, one can see that the accounts that are often mentioned had very close relationship with the candidates during the data collection period.

On the other side of the directed edges, however, are active users who often mention the candidates. Some of them are accounts to openly support the candidates, such as 'Feel the Bern!' for the Sanders network and 'Go Hillary 16' for the Clinton network. Others, on the other hand, are avid Twitter users with high level of activity.

3.3 Reply Network

Reply networks tend to be the smallest out of the three networks. From Table 3, one can observe that the number of reply edges are less than 10% of retweet or mention edges. This may be due to the fact that replies portray people's response to pre-existing tweets. This puts the person into a more active and interactive spot, where the user is expressing an opinion in response to a particular event or idea. On the contrary, one can freely mention various users in any tweet and easily retweet others' statuses without the creative process. Hence, reply network might be the most opinionated network out of all three networks.

Upon looking at the data (appendix A.2), one may observe that most of the users who were replied by others the most were the candidates themselves or the major media companies. Two users that stand out are 'Megyn Kelly' and 'issa' in Trump network. Megyn Kelly is a Fox News anchor who has been clashing with Donald Trump, and she co-moderated Republican presidential debate on January 28th which Trump boycotted [18]. 'issa' is a popular Twitter/Youtube user with more than 700,000 followers on Twitter and 2 million subscribers on Youtube; he had openly criticized Trump in January. As such, all the figures that got most replies are closely connected to the candidates.

On the other hand, just as in the mention network, the users who actually reply to these popular figures are individual users. Yet, some of these accounts portray strange above-average activity. For example, an account 'PetitionForChange' replied 137 tweets regarding Trump in the span of 4 days, while the total number of tweets it had ever written was 170. Hence, one may guess that some of these accounts may be temporary accounts used for specific goal.

3.4 Retweet Network

Retweet functionality, which allows reposting someone else's tweet with a click of a button, is an easy way to reproduce information one is interested in and broadcast to one's own network. Indeed, one can observe from the dataset that more than 50% of all posted tweets are retweets. For such reason, retweet networks are often studied to measure the spread of information. Each tweet by itself does not contain much information for it is a simple reproduction of the original tweet. However, the network created and the story it tells is not trivial.

A natural question that might arise is how often a tweet is retweeted. Undoubtedly, a tweet from a year ago will likely not be retweeted as often as a tweet from a day ago. It is worth noting, however, that there are occasions when old tweets resurface and gets retweeted.

One can observe from Figure 7 that there are cases when very old tweets are retweeted. These are some of the examples :

- Text : RT @Market_JP: Hillary Clinton said Monday she is the only candidate who would exercise the leadership needed to end the war in Iraq [Time of original tweet : 2008/03/18]
- Text : RT @realDonaldTrump: Be sure to tune in and watch Donald Trump on Late Night with David Letterman as he presents the Top Ten List tonight! [Time of original tweet : 2009/05/04]

Considering that Twitter started its service back on summer of 2006 [19], these are very old tweets. Indeed, it is almost remarkable that these tweets were found and retweeted. Excluding these special cases, one can observe from log-log graph in Figure 8 that most retweets occur within less than a day, and the frequency quickly decreases. An interesting characteristic of this graph is that it also seems to somewhat follow power-law with a fat tail, with the value of γ about 1.5.

The basic statistics regarding retweet time is summarized in Table 4. One noteworthy measure is the median: we see that about 50 % of the retweets are done within an hour of the original tweet. This number emphasizes

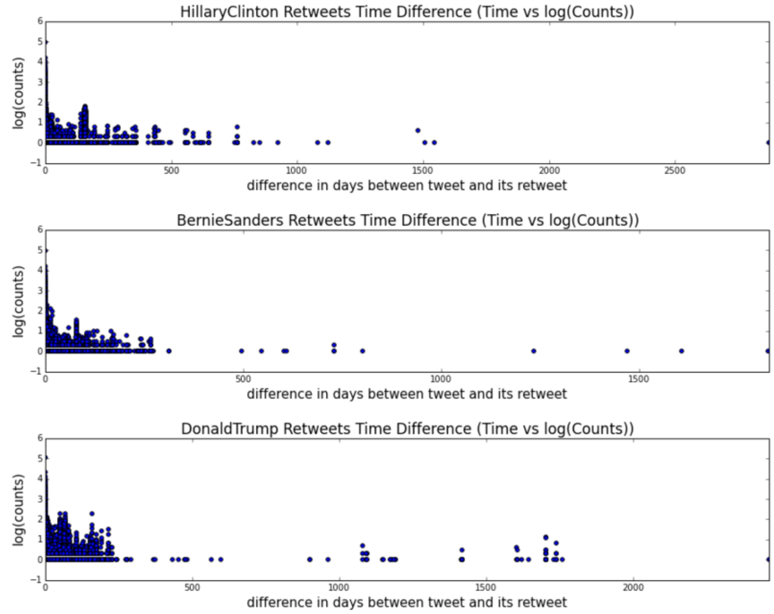

Figure 7: Distribution of time difference between original tweet and its retweet of each network (day vs log(counts)).

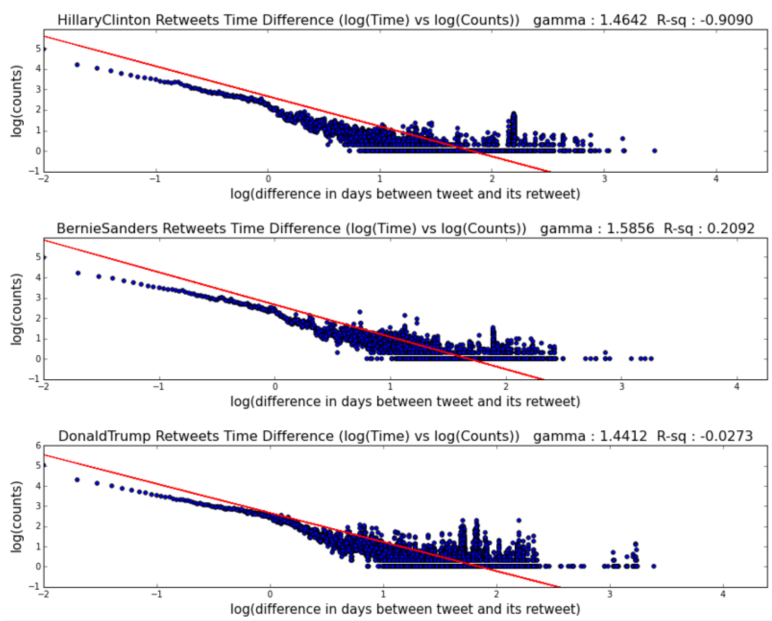

Figure 8: Distribution of of time difference between original tweet and its retweet of each network (log(day) vs log(counts)). Linear regression was fit using Clauset et al.'s power-law distribution fitting algorithm.

	Retweet Time			
Candidate	Min (seconds)	Mean (days)	Median (minutes)	Max (days)
Hillary Clinton	0	5.9848	48.88	2872
Bernie Sanders	0	1.7311	55.88	1826
Donald Trump	0	4.7509	67.45	2460

Table 4: Retweet time statistics.

how rapidly information spreads and disappears on Twitter. We also observe that the minimum retweet time is 0 seconds for all the networks. How is this possible? Consider the following 0-second retweets : ‘

- Hillary Clinton
  - tweet : ‘RT @BreakingNews: State Dept. confirms it will withhold 7 Clinton email chains; notes documents not classified at time they were sent’
  - Original tweeter ID : ‘Breaking News’
  - retweeter ID : ‘Breaking Politics’
- Bernie Sanders
  - tweet : ‘RT @pmarca: “I personally happen not to be a great believer in the free enterprise system for many reasons.’
  - Original tweeter ID : ‘pmarca’
  - retweeter ID : ‘rtpmarca’
- Donald Trump
  - tweet : ‘RT @voxdotcom: 3 charts show how Donald Trump dominated the Fox News debate by skipping it’
  - Original tweeter ID : ‘Vox’
  - retweeter ID : ‘Una Vox’

One can easily observe that the original tweeter and retweeter are closely related. For Hillary Clinton and Donald Trump, the retweeter account seems to be a child account of the original account. For Bernie Sanders, the retweeter account name explicitly states that it is for retweeting original ID. Hence, one can assume that there are cases in Twitter where some accounts automatically retweet others’, possibly to increase awareness and spread the information. Indeed, it is a common behavior used by various companies, and upon searching ‘automating retweets’ in search engines, one can find numerous tutorials on how to set-up automated retweets.

3.5 Combined Network

Lastly, we can also combine all the above networks into one. We simply count the edges from each network with equal importance. However, if needed, one may also combine networks by giving different weights to each network. For example, if deemed appropriate, reply network may have more edge weight than those of retweet or mention networks. This may be a reasonable choice, especially if one considers the fact that the size of the reply network is much smaller than the other two.

By combining networks we probably lose as well as gain information. We lose information about peculiar characteristics for individual networks by merging the datasets. However, as we have a bigger dataset we may be able to gain information with regards to the big picture of the network. Even though one may apply different network analyses for each of the three different networks, the work becomes repetitive and the result likely not so significantly different. Hence, from this point and on, we will be using the combined network for our analysis.

4 Sentiment Analysis

Sentiment analysis is the evaluation of a sentence to determine whether the opinion portrayed in the writing is positive or negative. A basic form of sentiment analysis computes Pointwise Mutual Information between words of phrases and sentiment keywords, then average over the sentence to get an estimate of the sentiment of the whole text [20]. First, we extract two-word phrases from reviews that match predefined patterns, such as ‘adjective+noun’ or ‘adjective+adjective‘. Denote the first word of the phrase as ‘$word_1$’ and the second as ‘$word_2$’. Then Pointwise Mutual Information (PMI) of these words is defined as :

$$PMI(word_1, word_2) = log_2\left(\frac{p(word_1 \& word_2)}{p(word_1)p(word_2)}\right) \tag{2}$$

where $p(word_1\&word_2)$ indicates the probability that the two words co-occur in the dataset. Note that here, $p(word_1)p(word_2)$ is the probability that two words co-occur if they are independent of each other. Hence, PMI measures the amount of information we gain about other word by acquiring presence of a word.

Then, Semantic Operation(SO) of a phrase is calculated as :

$$SO(phrase) = \sum_{keywords} s(keyword)PMI(phrase, keyword) \tag{3}$$

where $s(keyword) = \begin{cases} 1 & \text{if keyword is positive sentiment} \\ 0 & \text{if keyword is negative sentiment} \end{cases}$. For example, consider a keyword set of {good, excellent, terrible}. Then $SO(phrase) = PMI(phrase, good) + PMI(phrase, excellent) - PMI(phrase, terrible)$.

However, one may easily see a problem in such an approach as it does not take into account the grammatical context. A simple negation of the positive sentence might still be evaluated positive for it includes many positive words. Hence, more sophisticated rules have been developed that takes into account such sentence structures.

Numerous sentiment analysis libraries trained on different datasets exist. A powerful example is Stanford NLP's (Natural Language Processing) deep learning for sentimental analysis, which constructs a representation of whole sentences based on the grammatical sentence structure, and utilizes neural network to predict the sentiment of the sentence accurately predicting 80% of the human-labeling of the sentences [21]. Another major toolkit on the field is NLTK (Natural Language Toolkit), a platform that provides numerous functionalities for python programs to work with human language data. Specifically, the sentiment analysis adaptation uses Word Sense Disambiguation, Naive Bayes and Maximum Entropy Classifier based on movie rating dataset to predict the sentiment of given texts. However, this library was very slow in processing the sentences, handling only about 10 tweets per minute. Other libraries that were considered include python labMT and UC Berkeley NLP library.

The library we use is VADER(Valence Aware Dictionary and sEntiment Reasoner), a lexicon and rule-based sentiment analysis tool that is tuned specifically for social media [22]. To construct the library, the developers have gathered and combined pre-known datasets of human sentiment, and added emoticons (*e.g. :-)*), acronyms, as well as initialisms(*e.g. LOL*). They then utilized Amazon Mechanical Turk(AMT), a crowdsourcing platform, to manually label the sentiment of each keyword. They designed a system where users would get bonus for correctly predicting the average label of the group, achieving highly accurate and low-standard deviation results. Then they constructed various grammatical and syntactical variations of sentences (e.g *'Yay. Another good phone interview'* → *'Yay!! Another GOOD phone interview!!!!'*)and rated them through AMT. Lastly, they gathered data across various domains including social media and movie reviews, and set human-validated ground truth regarding sentiment intensity. The developers of the reasoner claims that VADER assessment of sentiment of tweets outperformed individual human raters, the former getting 0.96 and the latter 0.84 F1 classification accuracy, respectively. The code was also computationally very fast, processing the whole dataset in less than a minute. Hence, we use this library to analyze the Twitter network of the candidates.

Example Sentence	**Measures**			
	Negative	Neutral	Positive	Compound
'I believe in Candidate'	0.0	1.0	0.0	0.0
'Candidate will be a great president'	0.0	0.494	0.506	0.6249
'Candidate spoke at Iowa today'	0	1.0	0	0.0
'Candidate ate lunch with people'	0	1.0	0	0.0
'I really do not want Candidate to be the president'	0.157	0.843	0.0	-0.1244
'I hate the Candidate; that person is very dishonest'	0.562	0.438	0.0	-0.8268

Table 5: Sentiment scores for sample sentences evaluated by VADER. Negative, neutral and positive indicate the probability that the sentence may be classified as respective sentiments. Compound gives an overall score between -1 and 1, 1 being very positive and -1 being very negative.

One can observe the performance of VADER on toy examples in Table 5. It computes a measure of how likely it is for the sentence to be negative, neutral, or positive respectively, and assigns scores (probability) for each class, the sum of three scores for one sentence being equal to 1. It also provides overall sentiment score between -1 and 1 in the column 'compound', -1 being very negative and 1 very positive.

One can observe that on the average, users portray neutral tendencies rather than polarized behaviors. This may be due to the fact that when people tweet about political candidates it is usually through an article on media.

	Measures			
Candidate	Negative	Neutral	Positive	Compound
Hillary Clinton	0.0578	0.8605	0.0817	0.0489
Bernie Sanders	0.0432	0.8699	0.0869	0.0962
Donald Trump	0.0635	0.8502	0.0864	0.0616

Table 6: Average sentiment per candidate. For each network, all the tweets were evaluated using VADER and then the scores were averaged.

Then their sentiment is portrayed in the context of the article, not in the tweet itself. Hence the tweets may be classified as neutral and media accounts are mentioned often across all the networks.

We attempt to classify each tweet negative, neutral, or positive by assigning it the class with the highest score in the result of VADER analysis. The result is summarized in Table 7. As we observed before, there are overwhelmingly many neutral tweets. Hence, we attempt binary classification, where we label a tweet to be positive if the positive score is higher than the negative score, and negative if the negative score is higher than the positive score. The result is summarized in Table 8. Overall, all networks have similar proportion of positive tweets, ranging between 19% and 21%. The number of negative tweets portray a wider range of values, with Bernie Sanders having the least and Donald Trump having the most. The numbers hint that the tweets regarding Donald Trump are more polarized than those of other networks. One may also assume that Bernie Sanders is receiving fewest criticisms, while Hillary Clinton is showing an average behavior of these two candidates.

	Measures		
Candidate	Negative	Neutral	Positive
Hillary Clinton	876	169,104	325
Bernie Sanders	259	178,402	551
Donald Trump	691	270,059	723

Table 7: Tri-class classification for each candidate. For each network, evaluated each tweet using VADER. Each tweet was labeled to be one of 'negative','neutral', or 'positive' by looking at the maximum of the three corresponding probabilities.

	Measures	
Candidate	Negative	Positive
Hillary Clinton	43,510 (11.53%)	72,179 (19.12%)
Bernie Sanders	35,784 (8.54%)	79,525 (18.97%)
Donald Trump	78,342 (14.89%)	112,731 (21.43%)

Table 8: Binary classification for each candidate. For each network, each tweet was evaluated using VADER. Then, each tweet was labeled to be 'negative' if negative score was higher than positive score, and 'positive' in the opposite case. Percentage indicates the proportion with respect to the total number of tweets of each network. Note that this labeling is not comprehensive since more than half of the tweets were actually neutral with 0 for both positive and negative probabilities.

5 Dataset Reduction

Given the size of the sparse matrix of the original dataset, it is computationally difficult to work with the network. Hence, we introduce a simple threshold to reduce the size of the data. We filter the edges by their overall counts and then we symmetrize the resulting network.

One can observe that about 30% to 40% of original edges had frequency of 1 and are removed when we set 1 as the threshold. After setting threshold equal 2, we only retain about 6% to 7% of the original data, obviously losing a lot of information. However, we may also view this as a feature selection, for we are selecting the edges that have

	Number of Edges			
Candidate	Original	Threshold = 1	Threshold = 2	Symmetrized
Hillary Clinton	295,761	222,249	20,631	20,511
Bernie Sanders	339,930	241,396	24,162	24,011
Donald Trump	413,056	308,514	24,258	24,130

Table 9: Reduced dataset. The edges are filtered by their counts; the edges are retained only if the weight (counts) is above the threshold. The resulting edges are then symmetrized.

counts of at least 3. This filtered dataset may be understood as the more significant portion of the whole network in that each user interacted with the other user via mention, retweet, or reply at least three times in less than a week.

Lastly, we symmetrize the network. If edges of both direction exist for a pair of nodes, we add the frequency of each direction and set it as the frequency of the undirected edge. In most cases, there exists only one direction of edge between a pair of nodes, as one can see from the small amount of reduction as a result of symmetrization. In such cases, the weight of the directed edge is used for the weight of the indirected edge.

Just as introducing threshold, symmetrizing the network removes a lot of information from the network. However, it helps us compute metrics that are unsuitable for directed networks.

6 Spectral Clustering

6.1 Background

A network may be analyzed in various ways. One of the standard methods is by attempting to understand the network in parts. This may take form of Principal Component Analysis (PCA) which reduces dimension of features, or clustering algorithms which attempts to group different parts of networks and make sense out of them.

One can easily think of various ways to form clusters within Twitter networks. There are numerous features for each tweet or user, including follower count, tweet count, tweet frequency, average retweet count, et cetera. Hence, one may be able to form hypotheses about different users of the network by clustering information about the users. However, we aim to understand how the candidate is received within its network, rather than focus on the individual users. Hence, we use sentiment analysis scores to determine the clusters of the network.

Various methods have been suggested to find the clusters of a network. A common example is community detection using modularity. The original algorithm proposed by Newman and Girvan has been studied, developed, and widely used to detect communities of networks [23]. It computes betweenness of all existing edges, removes the edge with the highest value, recalculates the betweenness values of other edges and repeats the process until no edges remain. Even though it is a simple algorithm that performs well, it is computationally expensive as it needs to recompute the betweenness of all the edges of every iteration. Moreover, our original adjacency matrix is so sparse that there is not much point in looking for separate communities using such a method. Hence, we utilize spectral clustering to obtain the desired clusters of the network.

6.2 Spectral Clustering for Directed Networks

The initial adjacency matrix we have, apart from being very sparse and large, is directed. Directed networks, unlike undirected networks, lack properties that allow easier and faster computation of metrics. Hence, spectral clustering is not often utilized for directed networks due to complex valued decompositions [24]. The topic is still being actively discussed, with researchers attempting to better understand the community structures of directed graphs. For the sake of computation, we decided to symmetrize the dataset and perform spectral clustering on symmetrized dataset.

6.3 Spectral Clustering on Symmetrized Dataset

We perform spectral clustering on the reduced and symmetrized dataset (as described in previous section) using the sentiment scores. It is important to note that we do not perform spectral clustering on the original adjacency matrix constructed by retweet, mention, on replies. Rather, we construct a new distance matrix by the sentiment scores of each user.

We get the unique users who appear in the reduced dataset created in section 5, and compute the average sentiment scores of their tweets. Note that for some users who were only either mentioned or replied to rather than tweeted something themselves, we cannot compute this measurement since we do not have any tweet data for them. After getting the three scores (negative, neutral, positive) we develop a weight matrix W which computes the pairwise Euclidean distance between scores of each user. We also construct a diagonal degree matrix D, whose (i, i) element is the degree of node i.

Then we construct the symmetric Laplacian L_{sym}:

$$L_{sym} = I - D^{-1/2} W D^{-1/2} \tag{4}$$

Next, we compute the largest k eigenvalues and corresponding eigenvectors of L_{sym}, where k is the desired number of clusters. After constructing the matrix of first k eigenvectors V, we normalize the row sums to become 1, forming matrix U.

$$V = [v_1, v_2 \ldots, v_k] \text{ where } v_1 \ldots v_k \text{ are column vectors that are first k eigenvectors of } L_{sym}. \tag{5}$$

$$U = \text{normalized matrix such that} u_{ij} = \frac{v_{ij}}{(\sum_t v_{it}^2)^{1/2}} \tag{6}$$

Then, we have a new representation of the data, where the jth row of U corresponds to the jth datapoint. To finalize our clustering, we use $k-$means to obtain the clusters for each point [25].

After obtaining the cluster labels for each dataset, we apply the results to the reduced-symmetrized adjacency matrix. Hence, we see actual interactions (retweet, mention, replies) between users who were classified by their sentiment scores.

6.4 Results

Unlike usual usages of spectral clustering, where we form clusters by similarity measure and try to understand the result, we already have some expected outcome. Since the distance matrix is developed by the sentiment scores, we expect to see a clustering of polarized crowd. That is, we would get neutral, positive, and negative crowd. Indeed, as one can observe in the appendix, users are clustered by their sentiment scores. When $k > 3$, we have numerous clusters that could be understood as neutral, and one each for positive and negative. One can view the results for the case $k = 5$ from Table 10, and the total result for different values of k in the appendix A.3.

Clustering Result		**Average Sentiment**		
Group	Proportion	Negative	Neutral	Positive
Hillary Clinton				
1	24.76%	0.0297	0.8914	0.0789
2	13.66%	0.2072	0.7208	0.0720
3	25.03%	0.1025	0.8584	0.0391
4	13.33%	0.0066	0.9879	0.0055
5	23.21%	0.0560	0.7933	0.1507
Bernie Sanders				
1	12.53%	0.0034	0.9926	0.0040
2	20.82%	0.0552	0.8984	0.0464
3	15.64%	0.0204	0.7818	0.1978
4	27.30%	0.1096	0.7962	0.0941
5	23.72%	0.0122	0.9016	0.0862
Donald Trump				
1	11.94%	0.1632	0.7949	0.0419
2	15.24%	0.0017	0.9959	0.0024
3	23.47%	0.0104	0.8920	0.0976
4	26.98%	0.0496	0.8619	0.0885
5	22.37%	0.0671	0.8174	0.1155

Table 10: Results of spectral clustering by sentiment scores of each network, with 5 clusters. One can observe that each cluster is either neutral, positive or negative. Results for different values of k can be found in the Appendix.

Then we use the reduced-symmetrized adjacency matrix as discussed before to construct the network with result of spectral clustering as the classes for each user. Again, we can not classify the users who never tweeted and simply received mentions or replies, and they are classified as a different class by themselves.

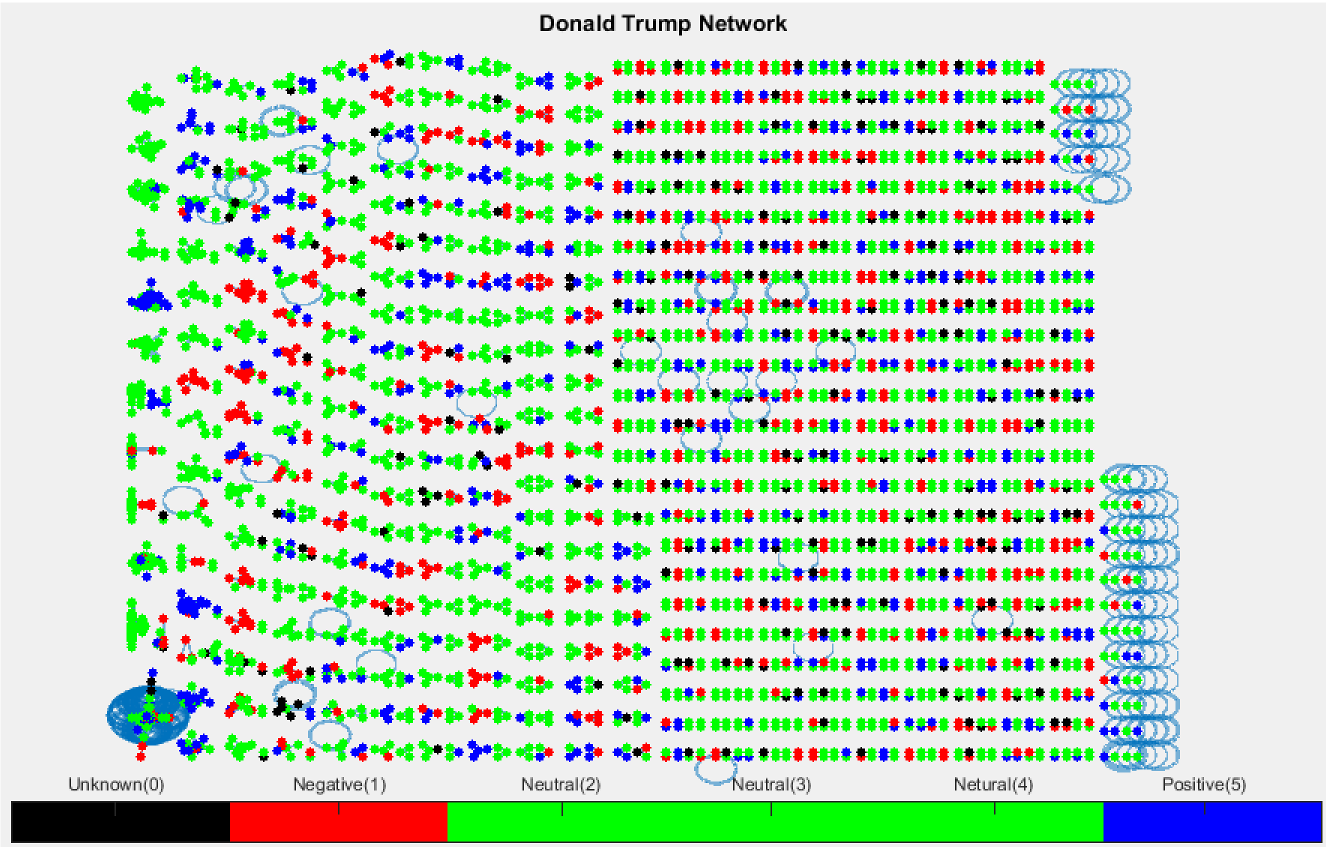

Figure 9: Graph for reduced and symmetrized dataset of Trump network (edge filtered by threshold of 2). Class was generated by performing spectral clustering on the sentiment scores and applied to the original adjacency matrix. Each color indicates the polarity of the cluster : Green(Neutral), Red(Negative), Blue(Positive), Black(Unknown).

A graph of such sparsity and size is very hard to visualize, and one often cannot gain any insight about the network from visualization of a network so big. However, this representation tells surprisingly many stories about the network. We specifically look at the Trump network, but similar observations can be made for other networks as well, which are included in the appendix.

- Self loops : One can easily observe numerous self loops on the right side of the graphs. The only self loops in the network are generated by replies. Hence, self loops are the cases when people are replying to their own comments, likely in conversation with another user. If any of the self-looped vertices is also connected to another vertex, there is a good chance that they are replying to the same tweet, doing a short interchange of ideas on the tweet.
- Motifs : A motif of a network is a subgraph that repeats itself throughout the network and show characteristics of the network. In the plot, the motifs with few nodes are apparent. Hence, we observe how individual users are talking about the topic in small groups.
- Greatest Connected Component : On the other hand, we also observe a great connected component on the left bottom corner. This component will very likely include the important users of the network as identified in section 3 and appendix.
- Sentiment Linkage : It is also visually noticeable that there is a tendency that nodes which are connected tend to portray a similar color. Even disregarding the fact that most nodes are green (neutral), one can often see groups of nodes identified as either negative or positive sentiment. This suggests that similar sentiment

is shared within the subgroups of this network. It may either be that they are retweeting each other, hence containing same text and therefore showing the same sentiment, or it may also be that they indeed share similar ideas. There were not that many clusters that were divided into blue/red, which would indicate a bi-polar crowd arguing.

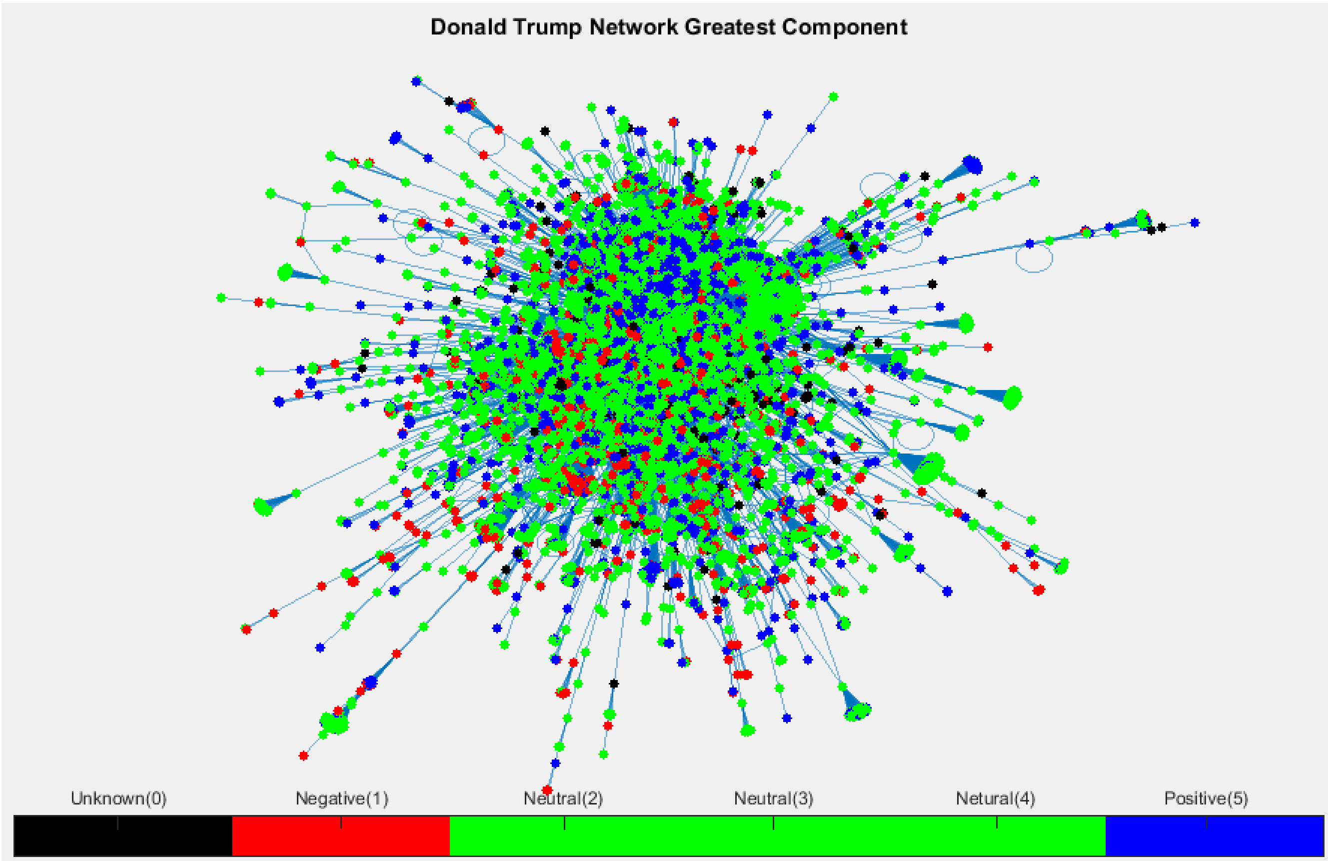

Figure 10: Graph of the greatest connected component of the Trump Network. Each color indicates the polarity of the cluster : Green(Neutral), Red(Negative), Blue(Positive), Black(Unknown).

We also take a brief look at the greatest connected component of the network. This component contains about 1/3 of the total nodes in original reduced network. Interestingly, one can see linear series of edges connecting the nodes near the outer parts of the network. It will require more analyses to determine whether the network bears the form of a star with a few hubs in the center as opposed to grid-like forms.

7 Word Distance

7.1 Background

Apart from sentiment analysis, another interesting approach one can take on the tweets is looking at the 'relevance' of words within them. The simplest representation of texts would be a text vector with each dimension indicating the presence of a word. One may choose words of interest depending on their significance, using metrics such as mutual information or frequency of appearance. Then one may vectorize the texts and build models that compute the relevancy between them. However, such a simple approach is limited in that 1. The result highly depends on the selected feature words 2. One cannot perform analyses other than classifying the text into groups. As much as that may be an interesting approach as well, we decided to look at the text from a slightly different direction.

Word2Vec is a group of models used to produce word-embeddings, recently developed by engineers at Google [26]. The basic model behind Word2Vec is skip-gram. Given a corpus of words w, their contexts c, and a corpus of text T, we aim to maximize the corpus probability :

$$\arg\max_{\theta} \prod_{w\in T} \left(\prod_{c\in C(w)} p(c|w;\theta) \right) \tag{7}$$

where $C(w)$ indicates the set of contexts for word w [27]. Given a text, we attempt to maximize the conditional probability that we accurately predict the words within some window of each word in the text. In other words, we attempt to build a model that are useful for predicting surrounding words in a sentence.

We model the conditional probability $p(c|w;\theta)$ using soft-max :

$$p(c|w;\theta) = \frac{e^{v_c \cdot v_w}}{\sum_{c'\in C} E e^{v_{c'} \cdot v_w}} \tag{8}$$

where v_c, v_w are vector representations for c and w where C is the set of all available contexts.

In the original paper, Mikolov et al' proposes two methods to compute this soft-max function: hierarchical softmax and negative sampling [26]. The implementation that we use utilizes the second method, which was proposesd to be more efficient. In this method, we attempt to maximize the probability that all the observations of (w, c) word-corpus pair came from observation of training data:

$$\arg\max_{\theta} \sum_{(w,c)\in D} logP(D = 1|w, c; \theta) \tag{9}$$

where $P(D = 1|w, c)$ denotes the probability that (w, c) came from the training data.

Substituting a softmax function, we obtain an objective function as follows :

$$\arg\max_{\theta} \sum_{(w,c)\in D} log \frac{1}{1 + e^{-v_e \cdot v_w}} \tag{10}$$

Since this has a trivial solution when $p(D = 1|w, c; \theta) = 1$ for all pairs of (w, c) we generate set D' of (w', c') pairs that do not exist in the training data, hence the name negative-sampling. Then the optimization becomes :

$$\arg\max_{\theta} \sum_{(w,c)\in D} log\sigma(v_e \cdot \ v_w) + \arg\max_{\theta} \sum_{(w,c)\in D'} log\sigma(v_e \cdot \ v_w) \tag{11}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$.

Lastly, a neural network is used to train the model. Note that it is not a deep neural network, but a shallow one of only two layers. Once the vectors are computed, one can utilize them to show remarkable results. Some examples are that vector('Paris')-vector('France')+vector('Italy') gives a vector close to vector('Rome') while vector('king')-vector('man')+vector('woman') is close to vector('queen') [28].

We use the java implementation of Word2Vec supplied by deeplearning4j.org, an open-source distributed deep-learning project [29], to analyze the dataset. We parse the text and remove common stop-words. However, parsing is not comprehensive, so that for some candidates, searching for similar words with just their last names return results while for others (e.g. 'Sanders' in both Clinton Trump network) there are no search results.

7.2 Results

We first compute the similarity scores between the candidates, both by their full name and then by their last name. The highest score that we observe between two candidates is 0.6818 between Clinton and Trump, while the lowest score we observe is −0.0881 between Sanders and Clinton in Trump network.

Then we compute the nearest words for each candidate in each network, the result shown in the appendix A.4. One can observe various words and symbols. Though they may seem random and disorganized at first, closer look into the words gives us few observations. For example, we will consider the Clinton Network :

- Candidates : Not surprisingly, the names of the candidates appear to be very close. For example, both Sanders and Trump appear in top 25 closest words of 'Hillary Clinton.'
- 'iowa' : The Iowa Caucuses were held on February 1st, 2016, only a few days after the tweet data was collected. Hence, it is not surprising that the word Iowa appears as a close word for both Clinton and Trump.
- 'rt' : As was summarized in Table 3, more than half of the tweets are retweets. All the retweets have 'rt' in the beginning of their text, and hence we see 'rt' to be close to almost all the candidates in each network.

Keywords	Network		
	Clinton	Sanders	Trump
HillaryClinton ↔ DonaldTrump	0.6818	0.3361	0.2565
HillaryClinton ↔ BernieSanders	0.1759	0.3441	0.1301
BernieSanders ↔ DonaldTrump	0.0862	0.2623	-0.0104
Clinton ↔ Trump	0.2390	0.1930	0.1299
Clinton ↔ Sanders	0.6684	0.0871	-0.0881
Sanders ↔ Trump	0.2439	0.1246	0.4428

Table 11: Similarity scores between candidates for each network

- ‘feminist’ : Feminism is a sensitive keyword in general, but especially in this presidential election. With the possibility of Clinton becoming the first woman major party runner for the upcoming election, some have argued whether it is an act of feminism to support Clinton or not. Without getting too much into this sensitive topic, it is still noteworthy that such keyword is ranked to be close to Bernie Sanders.
- ‘tired’, ‘top’, ‘disdain’, ‘ex-top’, ‘right’, ‘aide’ : These words seem like a random collection, but they make sense once we consider the title of an article published on January 25th : ‘Trump Proven Right: Ex Top Hillary Clinton Aide: ‘She’s’ Tired; Disdains Meeting Voters.’ [30] Trump had criticized Clinton for lacking the stamina for the campaign and presidency, and the article had commented that Clinton’s previous campaign manager confirmed such a statement. As these words appear to be very close to Trump’s name in Clinton network, we can see that this article was widely tweeted about.

8 Concluding Remarks

8.1 Remarks

We observe various interesting characteristics of Twitter networks of the three candidates of the 2016 U.S. Presidential Election. After collecting the data, network for each candidate is constructed with users as nodes and edges connecting them if one retweeted, replied to, or mentioned another user.

Looking at the significant users of each network, we confirm that the candidates or the media have the highest indegrees, while individual users or accounts that support the candidates have the highest outdegrees. We observe the power-law behavior of the networks, especially that of the outdegree distribution. Interestingly, the retweet distribution seems to roughly follow the power-law as well. We look at the cases when retweets happen within a fraction of time, and verify that they are automated procedures. Then we utilize the VADER library to perform sentiment analysis of the tweets. Most of them are identified as neutral; hence we attempt to do binary classification with positive and negative score, obtaining a rough classification that hints at the average sentiment of each candidate’s network. Next, to perform spectral clustering we reduce the size of the networks by applying the threshold of 2 for each edge weights. We construct a distance matrix by the user sentiments, and utilize spectral clustering to obtain classes of each user. Then we apply that classification to the reduced network, and make interesting observations regarding the network. Lastly, we use word2vec and compute the similarity between the words, again gaining insight about the network.

8.2 Future work

Various analyses have been done on the dataset. However, there are numerous other work that can be done. An example is cleaning the text and then utilizing the result of Word2Vec. As it gives a powerful representation of the words of the networks, we may be able to gain interesting results by vector computations on cleaner result. We can also try co-clustering on the dataset, forming clusters of sentiment as well as the actual connections.

Access to larger dataset across longer periods of time may also give us insight about a change in people’s opinions on the candidates. For this paper, we consider a fixed time window and do not consider any changes. Hence, it would be interesting to see if one can identify changes in people’s opinions on the candidates and how they align with the election results. Moreover, in generation of the network we do not consider the order of the tweets. That is, if we are interested in the information spread, it might not be so reasonable to add edges $A \to B$ and $B \to C$ if

A retweeted B before B retweeted C. Therefore, taking into account the order of the edges being added may be crucial in network construction.

Lastly, we can perform more analyses on the greatest connected component. We can look at the overall structure of the network as well as perform clustering algorithms for just the greatest connected component. As we state in the beginning of the paper, each network is unique and may be able to tell interesting, different stories. And that may hold for subnetworks as well.

We hope to return to this topic as the election progresses, and further identify the connection between the current events and the social media.

References

[1] Andrew Perrin *“Social Media Usage : 2005-2015”*, “PewResearch Center”, October 8,2015.

[2] Josh James *“Data never sleeps 2.0”*, Domo, Inc. April 23, 2014.

[3] Mike Isaac *Stoked by mobile ads, Facebook builds up steam,* New York Times, Jan 27, 2016.

[4] *Twitter Q3 2015 Company Metrics p.1*, October 2015.

[5] Twitter, Inc. *Developers : Overview - Case Studies*, https://dev.twitter.com/overview/case-studies, accessed Jan 20, 2015

[6] Caleb Davis, et al. *How Does News Diffuse Through Twitter*, Arizona State University 2013.

[7] Tauhid R. Zaman, et al. *Predicting Information Spreading in Twitter*, Microsoft Research 2010.

[8] Kristina Lerman *Dynamics of information spread on networks*, USC Information Sciences Institute.

[9] Marc A. Smith et al. *Mapping Twitter Topic Networks : From Polarized Crowds to Community Clusters*, PewResearch Center, February 2014.

[10] Wilson Andrews et al. *“Who is Running for President”*, The New York Times. Feb 20, 2016.

[11] *The Impact of Social Media on the 2016 Election*, http://www.intermarkets.net/blog/the-impact-of-social-media-on-the-2016-election/ October 8, 2015.

[12] Twitter, Inc. *API Rate Limits*, https://dev.twitter.com/rest/public/rate-limiting, accessed Jan 27, 2016.

[13] Twitter, Inc. *Developer Agreement & Policy*, https://dev.twitter.com/overview/terms/agreement-and-policy, May 18, 2015.

[14] Twitter, Inc. *The Search API*, https://dev.twitter.com/rest/public/search, accessed Jan 27, 2016.

[15] Albert-Laszlo Barabasi and Eric Bonabeau *“Scale-Free Networks”*, Scientific American. May 2003.

[16] Y. Virkar, A. Clauset *’Power-law distributions in binned empirical data’* Annals of Applied Statistics 8(1), 89 - 119 (2014).

[17] Daniel Kreps *“Watch Bernie Sanders Sing with Vampire Weekend at Iowa Rally”* Rolling Stone, January 31st, 2016.

[18] Callum Borchers *“The Long, Strange History of Donald Trump-Megyn Kelley Feud* The Washington Post, January 27th, 2016.

[19] Nicholas Carlson *The Real History of Twitter* Business Insider, April 13, 2011.

[20] Peter D. Turney *“Thumbs Up or Down? Semantic Orientation Applied to Unsupervised Classification of Reviews”* Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002.

[21] Richard Socher et al. *“Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank”* Stanford University, 2013.

[22] C. J. Hutto and Eric Gilbert *“VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text.”* Eighth International Conference on Weblogs and Social Media(ICWSM-14), June 2014.

[23] M. E. J. Newman, M. Girvan *’Finding and evaluating community structure in networks’* Physical Review E. 69, February 26th, 2004.

[24] Yuemeng Li et al. *(*“Analysis of Spectral Space Properties of Directed Graphs using Matrix Perturbation Theory with Application in Graph Partition”) 2015 IEEE International Conference on Data Mining, 2015.

[25] Ulrike von Luxburg *A Tutorial on Spectral Clustering* Stat Comput (2007) 17:395-416, August 22nd, 2007.

[26] Tomas Mikolov, et al. *’Efficient Estimation of Word Representations in Vector Space’*. CoRR, abs/1301.3781, 2013.

[27] Yoav Goldberg, Omer Levy. *'word2vec Explained: Deriving Mikolov et alś Negative-Sampling Word-Embedding Method'*. CoRR, abs/1402.3722, February 14th, 2014.

[28] Google, Inc. *'word2vec project information'*. https://code.google.com/archive/p/word2vec/, accessed March 5th, 2016.

[29] Deeplearning4j Development Team (). Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. http://deeplearning4j.org

[30] Kristinn Taylor. *'Trump Proven Right: Ex Top Clinton Aide: 'She's Tired'; Disdains Meeting Voters'*. Gateway Pundit, January 25th, 2016.

A Appendix

A.1 Degree Distribution

A.1.1 Hillary Clinton

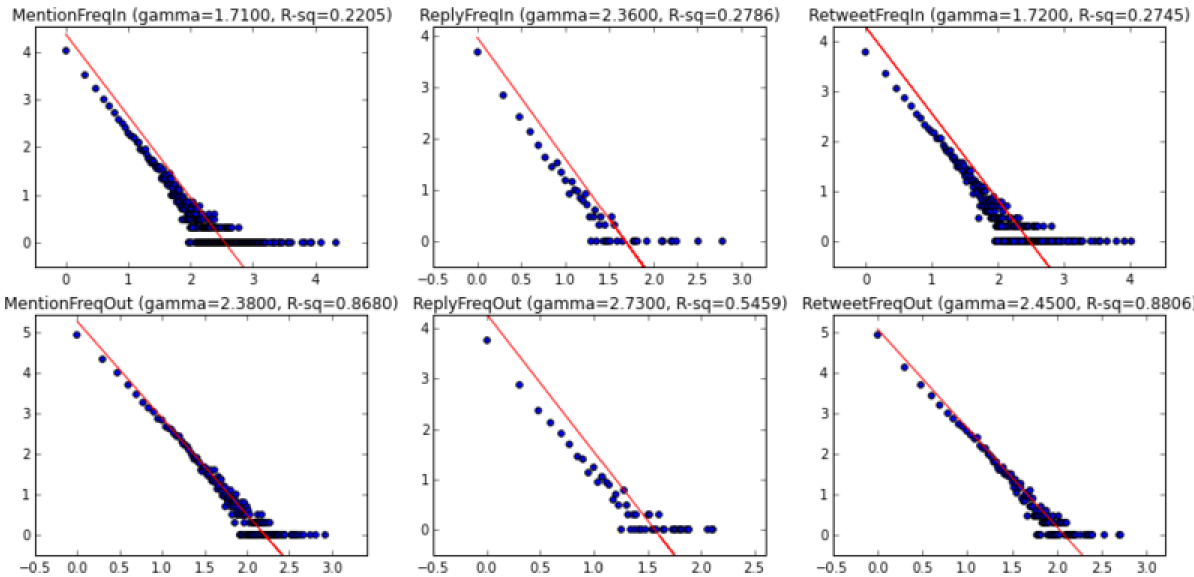

Figure 11: In/Out-degree distribution of each subnetwork of Hillary Clinton. (log-log scale). Linear regression was fit using Clauset et al.'s power-law distribution fitting algorithm.

A.1.2 Bernie Sanders

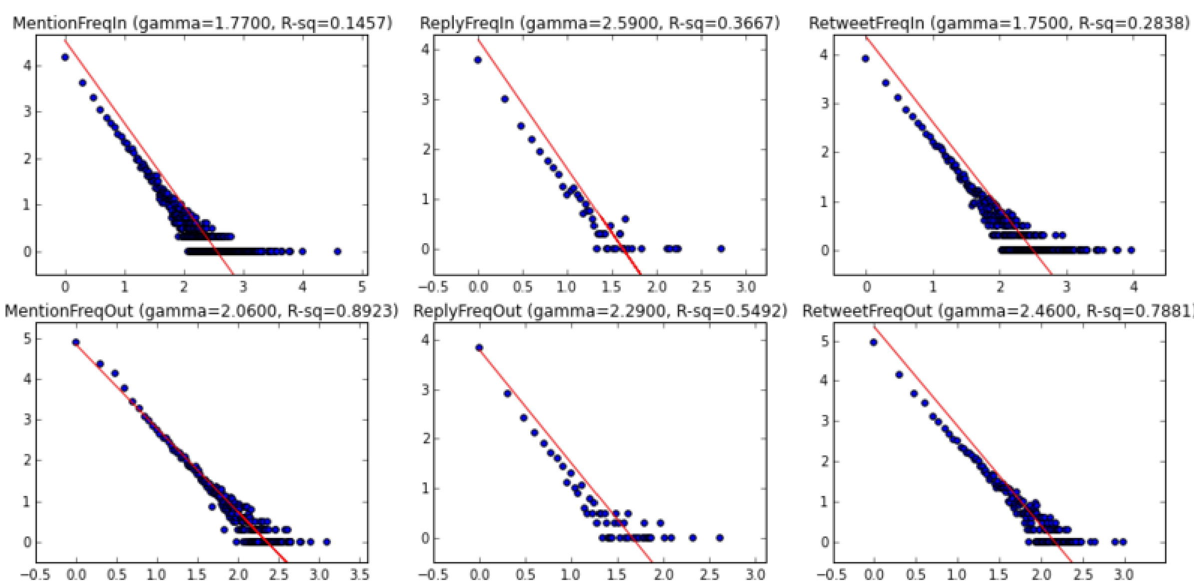

Figure 12: In/Out-degree distribution of each subnetwork of Bernie Sanders. (log-log scale). Linear regression was fit using Clauset et al.'s power-law distribution fitting algorithm.

A.1.3 Donald Trump

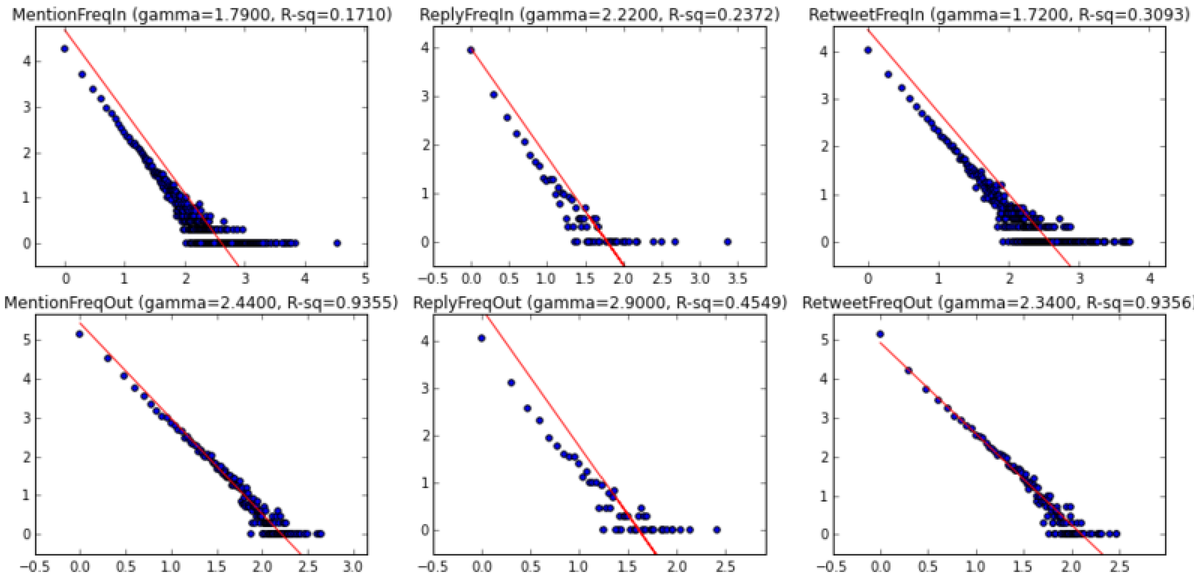

Figure 13: In/Out-degree distribution of each subnetwork of Donald Trump. (log-log scale). Linear regression was fit using Clauset et al.'s power-law distribution fitting algorithm.

A.2 Significant Users of Each Network

A.2.1 Hillary Clinton

	Freq	Name	Follower Count	Status Count	Friend Count
		Mention (In)			
1	21493	Hillary Clinton	5289503	3849	573
2	12306	Donald J. Trump	5947029	30585	50
3	8589	Bernie Sanders	1304656	6210	1436
4	8390	NathanZed	142408	24860	1026
5	6855	The New York Times	24066206	217581	986
		Mention (Out)			
1	821	George Fayner	562	20015	1049
2	655	Ron Glass	63	1604	446
3	560	Go Hillary 16	326	19972	84
4	458	Daniel John Sobieski	40146	807396	20348
5	430	Daniel John Sobieski	18957	468364	8204
		Reply (In)			
1	606	Hillary Clinton	5289503	3849	573
2	319	Donald J. Trump	5947029	30585	50
3	183	Fox News	7921421	231178	394
4	163	CNN	22919864	76920	1112
5	159	The New York Times	24066206	217581	986
		Reply (Out)			
1	129	Lincoln Bevers	40	758	110
2	125	nick shane	3	642	4
3	114	theblazingtruth	286	138194	342
4	77	Carl Nyberg	1903	57038	2023
5	76	GAPeach	6072	150809	950
		Retweet (In)			
1	10897	Donald J. Trump	5947029	30585	50
2	8383	NathanZed	142408	24860	1026
3	6284	Stefan	357	5429	443
4	4485	Hillary Clinton	5289503	3849	573
5	3653	David Sirota	41617	45196	2468
		Retweet (Out)			
1	503	George Fayner	562	20015	1049
2	494	Ron Glass	63	1604	446
3	342	Go Hillary 16	326	19972	84
4	249	Marcy McGowan	22148	672498	24356
5	243	Damon Bethea	3304	112442	3030

Table 12: Significant users of Hillary Clinton Network. The users were ordered by their in/out degree within each subnetwork, and five users with the greatest degrees are displayed.

A.2.2 Bernie Sanders

	Freq	Name	Follower Count	Status Count	Friend Count
		Mention (In)			
1	39888	Bernie Sanders	1322660	6242	1438
2	10061	CNN	22918985	76916	1112
3	6321	YouTube			
4	5981	Bernie's Spotify	1566	59	3
5	4568	Vampire Weekend			
		Mention (Out)			
1	1242	Feel the Bern!	738	27437	74
2	778	AMERiCA_D0N'T_GET	933	25811	3200
3	612	Aerin Cruz	1575	84497	1290
4	572	Anne Absalom	961	14678	2236
5	451	Hedge_Shot	1940	34258	2295
		Reply (In)			
1	535	Bernie Sanders	1322660	6242	1438
2	170	Hillary Clinton	5289503	3849	573
3	160	Donald J. Trump	5939329	30581	49
4	141	Bernie Sanders	1317634	13582	1909
5	129	CNN	22918985	76916	1112
		Reply (Out)			
1	421	WorldTaiChiDay	559	27258	20
2	209	Smooch Mcgee	62	913	103
3	123	Michael Martin	128	2611	270
4	102	Nigel Clarke	250	2217	65
5	92	Tony	92	5888	137
		Retweet (In)			
1	9417	CNN	22918985	76916	1112
2	5975	Bernie's Spotify	1566	59	3
3	5578	Bernie Sanders	1322660	6242	1438
4	3505	Bernie Sanders	1913	257	2000
5	3024	Viral Buzz News	103350	11649	23244
		Retweet (Out)			
1	978	Feel the Bern!	738	27437	74
2	778	AMERiCA_D0N'T_GET	933	25811	3200
3	446	Aerin Cruz	1575	84497	1290
4	397	Anne Absalom	961	14678	2236
5	303	Tara Lyman	1123	72559	561

Table 13: Significant users of Bernie Sanders Network. The users were ordered by their in/out degree within each subnetwork, and five users with the greatest degrees are displayed.

A.2.3 Donald Trump

	Freq	Name	Follower Count	Status Count	Friend Count
Mention (In)					
1	34946	Donald J. Trump	6038974	30629	49
2	7078	CNN	22918985	77056	1112
3	6089	YouTube			
4	5882	The New York Times	24117870	217782	986
5	5615	TIME.com	9572944	157010	844
Mention (Out)					
1	452	Philip Monaco	1878	42384	3099
2	436	Denise Martin	1003	7272	2450
3	399	Jim	917	66454	2088
4	321	Magma Taishi	1144	35186	2070
5	301	Donald Trump is...	84	2333	2
Reply (In)					
1	2414	Donald J. Trump	6038974	30629	49
2	488	Fox News	7922048	232287	394
3	318	Megyn Kelly	1463635	8307	1202
4	246	CNN	22918985	77056	1112
5	155	issa	689288	9199	15239
Reply (Out)					
1	267	Denise Martin	1003	7272	2450
2	137	PetitionForChange	0	170	3
3	111	Progress and Prosper	2274	3326	1990
4	92	missing in action	28	1387	240
5	91	Canadians4Trump	1613	11687	4994
Retweet (In)					
1	5251	TIME.com	9572944	157010	844
2	5174	khendra	22986	32586	17771
3	4862	CNN	22918985	77056	1112
4	4757	The New York Times	24117870	217782	986
5	4520	issa	689288	9199	15239
Retweet (Out)					
1	299	Philip Monaco	1878	42384	3099
2	292	Jim	917	66454	2088
3	247	Magma Taishi	1144	35186	2070
4	204	Donald Trump is...	84	2333	2
5	202	Robert Hochu	437	14255	204

Table 14: Significant users of Donald Trump Network. The users were ordered by their in/out degree within each subnetwork, and five users with the greatest degrees are displayed.

A.3 Spectral Clustering Result

A.3.1 Hillary Clinton

Clustering Result		Average Sentiment		
Group	Proportion	Negative	Neutral	Positive
2 Clusters				
1	49.54%	0.0511	0.9124	0.0365
2	50.46%	0.0989	0.7885	0.1126
3 Clusters				
1	37.04%	0.0847	0.8495	0.0657
2	26.92%	0.0279	0.9498	0.0223
3	36.04%	0.1008	0.7757	0.1236
4 Clusters				
1	30.00%	0.0175	0.9420	0.0405
2	15.24%	0.2010	0.7246	0.0744
3	24.02%	0.0385	0.8105	0.1510
4	30.74%	0.0979	0.8530	0.0491
5 Clusters				
1	24.76%	0.0297	0.8914	0.0789
2	13.66%	0.2072	0.7208	0.0720
3	25.03%	0.1025	0.8584	0.0391
4	13.33%	0.0066	0.9879	0.0055
5	23.21%	0.0560	0.7933	0.1507

Table 15: Results of clustering by sentiment scores of the Clinton network.

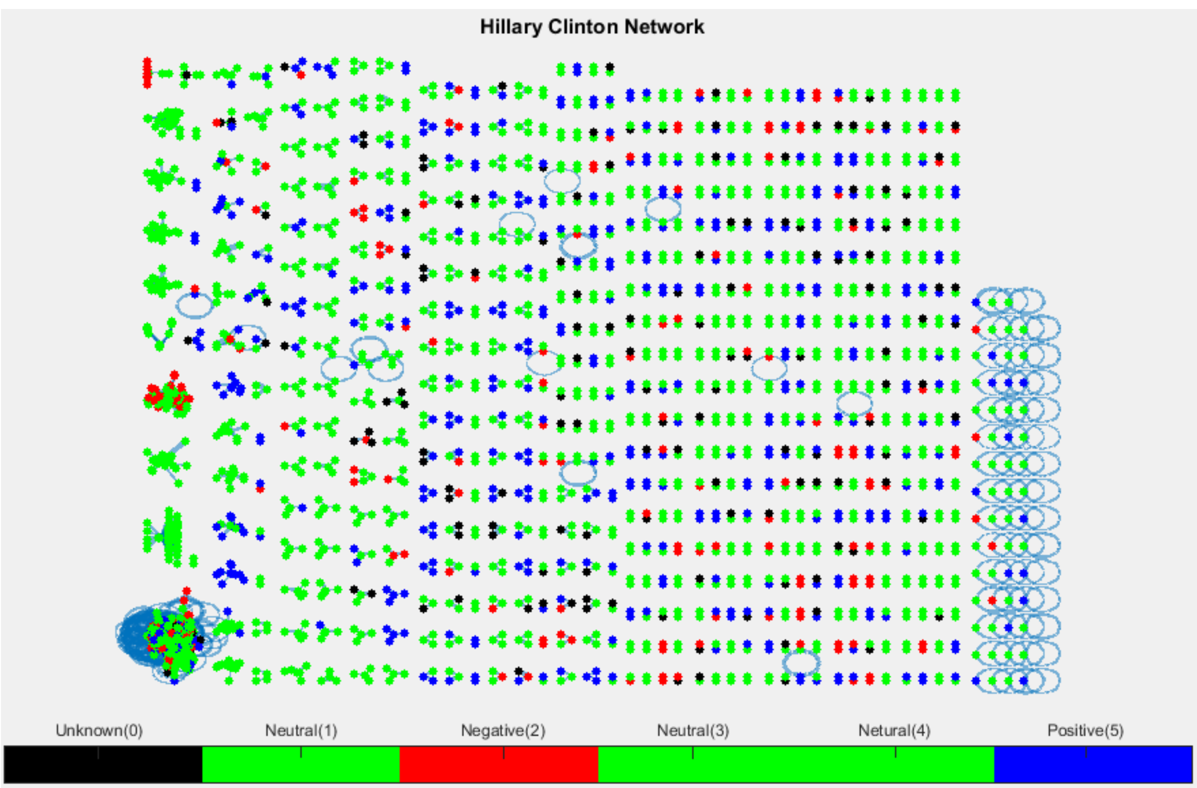

Figure 14: Graph of the reduced-symmetrized Clinton network after applying the classes obtained from spectral clustering of sentiment scores of each user.

A.3.2 Bernie Sanders

Clustering Result		Average Sentiment		
Group	Proportion	Negative	Neutral	Positive
2 Clusters				
1	48.09%	0.0311	0.8345	0.1344
2	51.91%	0.0636	0.8929	0.0435
3 Clusters				
1	34.61%	0.1024	0.8207	0.0769
2	29.43%	0.0163	0.9557	0.0280
3	35.96%	0.0215	0.8329	0.1456
4 Clusters				
1	25.34%	0.0167	0.9061	0.0772
2	19.20%	0.0185	0.9692	0.0123
3	31.41%	0.1067	0.8179	0.0754
4	24.05%	0.0275	0.7994	0.1731
5 Clusters				
1	12.53%	0.0034	0.9926	0.0040
2	20.82%	0.0552	0.8984	0.0464
3	15.64%	0.0204	0.7818	0.1978
4	27.30%	0.1096	0.7962	0.0941
5	23.72%	0.0122	0.9016	0.0862

Table 16: Results of spectral clustering by sentiment scores of the Sanders network.

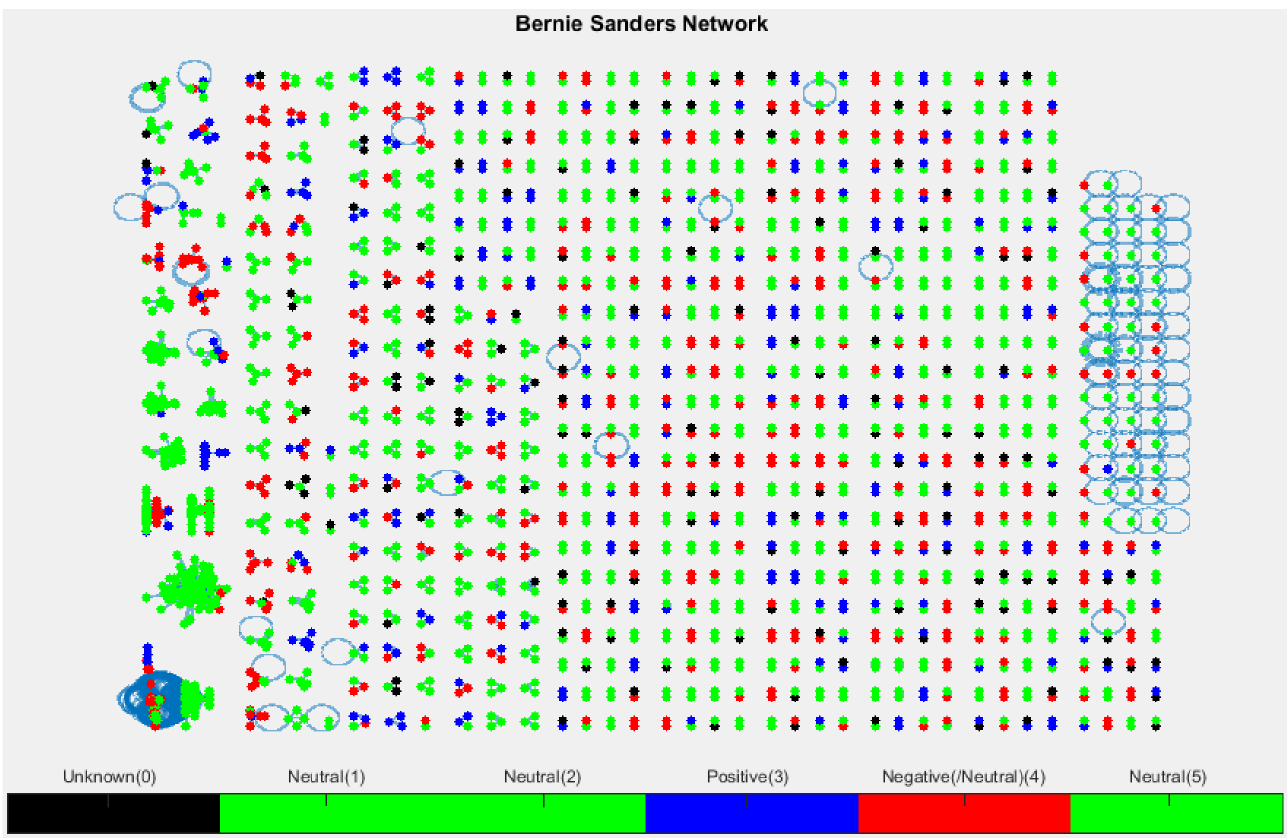

Figure 15: Graph of the reduced-symmetrized Sanders network after applying the classes obtained from spectral clustering of sentiment scores of each user.

A.3.3 Donald Trump

Clustering Result		Average Sentiment		
Group	Proportion	Negative	Neutral	Positive
2 Clusters				
1	52.11%	0.0654	0.9036	0.0311
2	47.89%	0.0345	0.8365	0.1291
3 Clusters				
1	25.14%	0.0088	0.9727	0.0184
2	37.45%	0.0283	0.8316	0.1401
3	37.41%	0.1009	0.8432	0.0559
4 Clusters				
1	28.66%	0.1077	0.8516	0.0408
2	26.26%	0.0177	0.9012	0.0811
3	16.30%	0.0040	0.9930	0.0029
4	28.78%	0.0500	0.7952	0.1547
5 Clusters				
1	11.94%	0.1632	0.7949	0.0419
2	15.24%	0.0017	0.9959	0.0024
3	23.47%	0.0104	0.8920	0.0976
4	26.98%	0.0496	0.8619	0.0885
5	22.37%	0.0671	0.8174	0.1155

Table 17: Results of spectral clustering by sentiment scores of the Trump network.

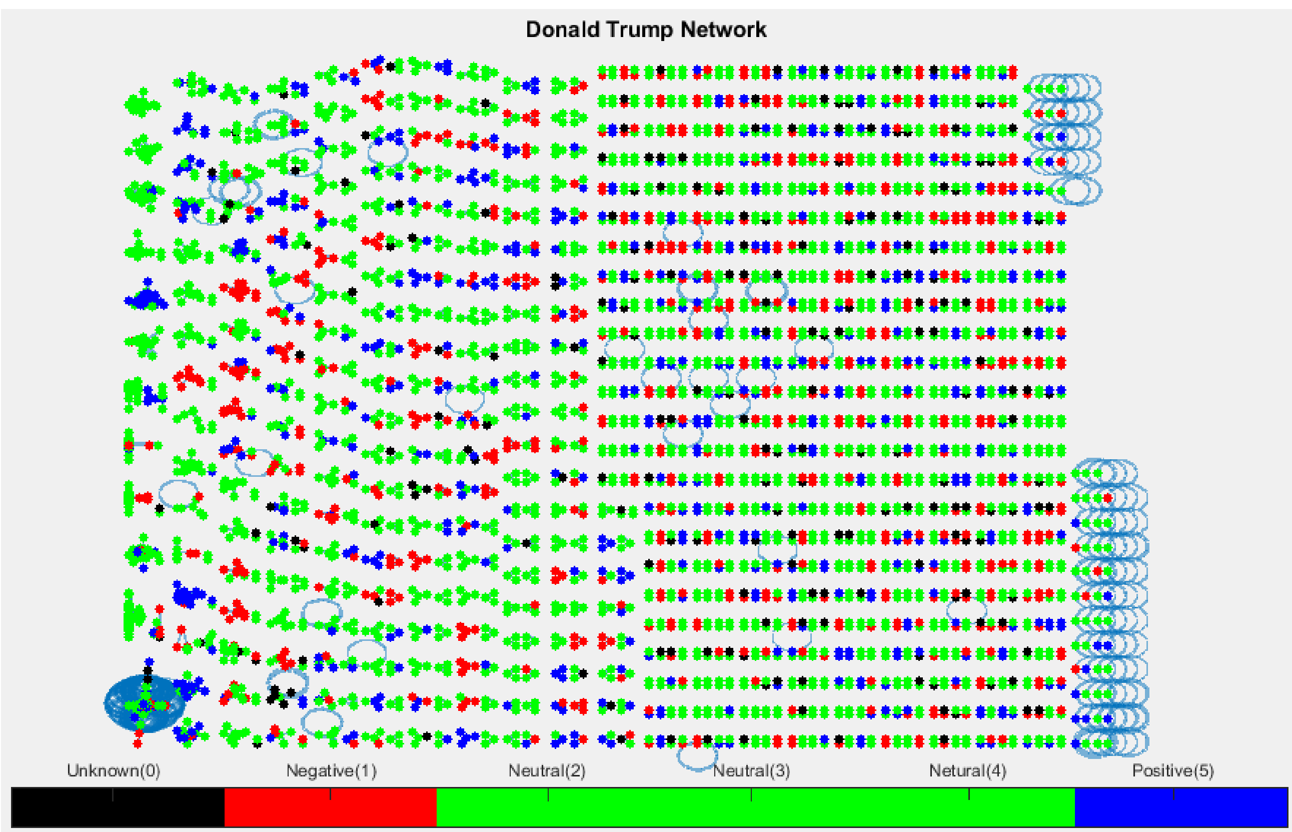

Figure 16: Graph of the reduced-symmetrized Trump network after applying the classes obtained from spectral clustering of sentiment scores of each user.

A.4 Word Similarity in Each Network

A.4.1 Hillary Clinton

Keywords	Top 25 Words With Highest Similarity
HillaryClinton	.., iowa, rt, obama, clinton, &, berniesander, hillary, donaldtrump, a, ”rt, bernie, like, in, one, i, say, -, dddd, the, call, #demtownhall, ”hillaryclinton, —, president
BernieSanders	sarandon„ gloria, @benshapiro:, #berningforlove, @sentinelsource, summer, fierstein, steinem„ https://t.co/wmml. . . , backyard, feminist„ david, recent, @nataliejose:, fictional, al, call, th, someone, harvey, neighbor?, fighter””, https://t.co/rdfxmpzcpr, larry, @lcvoter
DonaldTrump	.., iowa, rt, obama, clinton, #hillaryclinton, hillaryclinton, berniesander, a, question, ”rt, in, like, one, i, real, -, people, the, call, #demtownhall, campaign, know, —, president
Clinton	.., no, iowa, rt, obama, media, #new, via, hillaryclinton, her, and, of, berniesander, hillary, a, donaldtrump, bernie, real, dddd, the, ”hillaryclinton, ask, ha, to, —
Sanders	
Trump	@americanlizzy:, @klsouth:, @s_t_o_p_terror:, tired’;, #trump, tired’;, https://t.co/dpzwdwkdmo, proven, #hillaryclinton, top, ”ex, https://t.co/qidqcpfdaw, https://t.co/dtdggddgrf, https://t.co/sbsadmlioi, right:, https://t.co/ukodwoqdgq, disdain, ex-top, https://t.co/cjtdynkhkd, ex, @stanleyecook:, meet, https://t.co/ddexkqdlxy, ‘she’, aide:

Table 18: Top 25 similar words for each keyword in the Hillary Network, computed using word2vec.

A.4.2 BernieSanders

Keywords	Top 25 Words With Highest Similarity
HillaryClinton	iowa, berniesanders', rt, obama, why, house, white, #berniesander, attack, democratic, talk, berniesander, @berniesander, bernie, "rt, d, say, -, dddd, call, #demtownhall, watch, sander, support, president
BernieSanders	rt, south, #scforbernie, https://t.co/mkvgcghzbu, he', read:, #berniesander, get, lawmaker, berniesander, covert, side, carolina, could, go, i, say, @ckcovertd:, #feelthebern, endorse, https://t. . . , guess, u, must, time
DonaldTrump	.., rt, https://t.co/qkoddbpdri, nur, dangerou, @cherguiambark:, https://t.co/ksmtsxukzt, would, get, horrible, berniesander, win, what', race, like, piss, dddd, @historicalpics:, twitter:, person, runn, campaign, he, endor. . . , president
Clinton	https://t.co/ddforirddt, big, poll:, cl.., gop, poll, leads:, dem, sande.., attack, #clinton, win, https://t.co/bcuhvlhjqd, obama':, cnn, race, berniesanders,.., remain, #fk,adviser:, #libuster, *really*, sander.., vs, primary
Sanders	uc-aft, https://t.co/dvwiykdkcn, #change, sanders", @jacarafish:, https://t.co/bddmydybcq, loon, socialist!, https://t.co/ydhbbsznbc, poem, https://t.co/fjpviljraq, @jillgirlddd:, #thislandisyourland, https://t.co/jesmdqddkd:, sand. . . , gto, https://t.co/qxdlrnjrbj, itsumo, rave, mlk, atlanta', mike:, sanders!, killer. . . , atlanta
Trump	donald, sketchy, https://t.co/adrddddfwd, https://t.co/dblkdktoor, despise, to.., fox, people"", #blizzardddd, total, is. . . ", still,, day!!!, ??, @redcricketblog:, likely', rat, "bernie, nuff, setup, ain't, asl, #flappyhillary, spurn, primary

Table 19: Top 25 similar words for each keyword in the Sanders Network, computed using word2vec.

A.4.3 Donald Trump

Keywords	Top 25 Words With Highest Similarity
HillaryClinton	@figdrewton:, delusion, islamophobia:, foundation, @woodruffbets), hilary, @forbes:, causes, share, near, five, value, fdn?, given, https://t.co/hydsdyvisd, health, https://t.co/cptayeqyem, https://t.co/anxdyijzsf, they're, foundation.", https://t.co/klmixdqdre, money, gave, care, https://t.co/mvviqsas...
BernieSanders	?lection, @henrydelesquen:, (?pisode, https://t.co/duhgygdfyd, laurence, enfer, r?ve, ever.", duel, pen, devant, ?lecteur, #primaire, beau..., inc, peu, c'est, @katmckinley:, ha?m, momma, ?lu, choisit, pa, d'un, ocupado
DonaldTrump	rt, republican, gop, he', fox, via, debate, get, "donaldtrump, new, in, like, ..., i, bill, o'reil, say, donaldtrump', -, #donaldtrump, the, t, cruz, time, —
Clinton	gumption""??, anything??, choose, best?, hilary, makin, lego, poll(, @mayoroflondon, president!, poll(d/dd-d/dd), @samouraiid:, bernie, maine:, @kateandr, reggie, hysteria, hell!, cash!, rt)!, afford, fourth, https://t.co/ddgtsdvkyd", $d,ddd, usa,
Sanders	
Trump	.., donald, republican, iowa,, for, gop, fox, debate, join, event, ral, on, new, "rt, in, #makeamericagreatagain, i, j, -, way, special, with, veteran, u, to

Table 20: Top 25 similar words for each keyword in the Trump Network, computed using word2vec.