The Polarization of Information on the Web

EE496 Final Report
Fall 2018

Department of Electrical Engineering
University of Hawaii

Charles Dickens

December 2018

Faculty Advisor: Professor Narayana Prasad Santhanam

Abstract

Many sources claim that the political climate is becoming increasingly polarized, but few attempt to quantify this feeling. Furthermore, the strict dichotomy of left vs. right is a simple and typically accepted narrative, but perhaps a more fine grained model which considers smaller schools of thought on a topic to topic basis may better capture what is truly happening on the web. With an objective understanding of the polarization of information on the web, researchers could pinpoint causes, and potentially come up with solutions to reverse this concerning trend and foster more fruitful conversation.

The primary focus of this project is to develop a system that can provide objective measurements of the polarity of the network of information and opinions on Twitter. The system will be developed with the intention to later include a variety of data types, such as news articles and other popular social network postings. We have implemented a semi-supervised approach to modelling the network of tweets with an un-directed weighted graph using a feedback loop which applies a state of the art clustering algorithm that was developed in house at UH Manoa's Big Data lab. Communities of the resulting network are identified and the polarization of the network is then quantified using graph conductance.

Our system has analyzed multiple topics of discussion on Twitter and has identified groups that align with our intuitions. The network structure including polarity calculations can be successfully measured and compared across topics. Future work on this project will be to incorporate more features into the methods for building and weighting the network of information.

Problem Statement and Project Objectives:

Popular microblogs such as Twitter have become primary sources of information for many citizens. Twitter encourages users to share their opinions and information about current issues and events, no matter how contrarian, and access to such a variety of viewpoints has the potential to foster awareness. However, it seems groups of society are forming increasingly polarized opinions on topics, leading to disagreements over even factual details. This concerning observation has been a common topic of conversation as of late, but an accepted method for quantifying the polarization between camps on a topic to topic basis has yet to be developed, leaving the dialogue and, as a result, the proposed solutions subjective and difficult to act upon.

The short term objectives of this study are

1. Develop a method for comparing the polarization of existing communities of thought at the topic specific level

2. Improve methods for modeling social networks

The contributions of this report are an implementation of a semi-supervised approach to building a social network model, and the framework for a workflow which models a network, finds community structure in the model, and measures the polarity using graph conductance.

Final Design

The final design of the system can be separated into four tasks: retrieving relevant information and opinions, modeling the network, identifying communities in the network model, and finally measuring the polarity. In depth explanations of each of the four tasks can be found in the *Data Collection and Analysis* section. The four tasks are handled by five different classes in the programmed implementation: TweetCollector, TweetFeatureExtractor, NetworkBuilder, Clusterer, and PolarityCalculator. The block diagram of the class structure for the system is seen in figure 4.

The TweetCollector class is responsible for retrieving relevant information and opinions. A TweetCollector instnace sets up a connection to the Twitter search API and then structures queries. TweetCollector makes the search and collects the results from the Twitter API and organizes the tweets into a .csv file.

The TweetFeatureExtractor class is a helper class for the other modules. A single instance of the of the class is shared among the NetworkBuilder, Clusterer, and Polarity Calculator class. This approach results in less repeated computation and uniform network node to tweet id mapping.

The NetworkBuilder and Clusterer class work together to both model the network and perform community detection analysis. The NetworkBuilder class initially assigns every tweet to is own community. Then a d dimensional vector for each tweet is calculated using the hashtags of the tweet where d is the number of communities. The d dimensional vector tells us how much the tweet is pointed in the direction of each of the d communities. Then the weight of the edge between each tweet is calculated using the cosine similarity metric. The network derived from the iteration is passed to the Clusterer instance where the backward path algorithm is applied to the network to identify a new set of communities. The backward path algorithm returns a hierarchical set of different potential clusterings. The clustering that is used is to continue the process is the coarsest clustering that separates tweets so that no group has tweets with contrasting sentiment. This process is repeated until convergence, that is until the Clusterer instance repeatedly returns the same clustering indicating that the process has reached a local optima.

Once the network model has been obtained and the communities of the network identified, the polarity calculations can be made. The polarity calculation task is handled by the PolarityCalculator class. The PolarityCalculator class uses the conductance metric to measure the polarity.

This design was tested and verified on a simulated dataset of 100 Tweets. A network was generated using an LFR model with a power law exponent for the degree distribution of 2, a power law exponent for the community size distribution of 2, a fraction of intra-community edges incident to each node of 0.8, a average degree of 10, and a minimum community size of 20. The nodes of the LFR model where then assigned hashtags using a distribution that is dependent on the community assignment of the tweet. Each community has a corresponding set of 5 hashtags that all share a sentiment value. Then, hashtags are assigned to tweets with probability of 0.95 if the tweet belongs to same community as the hashtag, or with probability 0.02 otherwise. The network communities are nearly perfectly retrieved from the clustering algorithm and the communities obtained do not contain contrasting polarity hashtags. The results of the simulated run can be visualize in the figure 1.

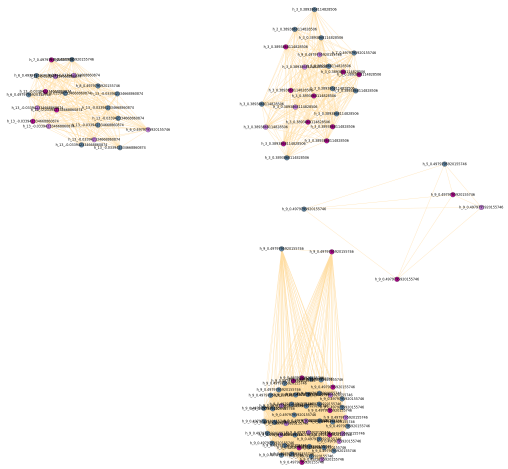


Figure 1: Simulated network using LFR benchmark and hashtag distribution based on bernoulli trials with probabilities dependent on the tweet's true community assignment.

Data Collection and Analysis

Retrieving Relevant Information and Opinions: The first task is to collect a sample of tweets which are disseminating information about a specific topic. The Twitter API provides a search data endpoint which can be queried for tweets within a specified time frame. This endpoint provides results that are uniformly sampled from the set of all Tweets that it deems is related to the query.

For this project, tweets regarding three topics, gun regulations, voting rights and turnout, and immigration policies, were collected. Twitter granted this project access the premium search API sandbox, which allows for 25000 tweets to be collected each month. The queries and results for the three topics are shown in the table 1.

Topic	Queries
Gun Control	'guns'
Voting	'voting rights', 'voter turnout', 'voter fraud' 'voter suppression', 'vote'
Immigration	'immigration', 'illegal immigrants', 'trump wall' 'border security', 'asylum seekers'

Table 1: Twitter API Search Queries by Topic

The tweets for gun control were collected first and a summary of the results can be seen in the table 2. The tweets for gun control were collected using the search query 'guns'. This query was used with the intention of being an unbiased search. However, At its current state, downstream analysis in the system soley considers the hashtags included in each tweet, thus a higher hashtag count is desired so that more connections within the network model can be made. Unfortunately, the query made to the search API resulted in only 500 tweets with hashtags out of 7500. Furthermore, not all of the tweets were actually related to the 'guns' we had in mind. It turned out that the band Guns-&-Roses was currently on tour and was a hot topic on twitter during the time frame of collection.

Topics	Count of Tweets w/ Hashtags	Count of Unique Hashtags
Gun Control	500	408
Voting	847	401
Immigration	356	187

Table 2: Twitter API Search Results by Topic

Based on experience with the twitter search engine and user behavior, a simple solution to the issue of low hashtag density is to make searches using trending keywords and phrases that lean towards one or the other side of the debate. However, meaningful results rely on the sample being a good representative of the population as a whole. To account for this, the two topics 'voting' and 'immigration', where searched for using queries that ranged across the spectrum of existing opinions on the topic based on our best intuition.

Modeling the Network: Once the relevant tweets are sampled, the next task is to build a weighted network modeling the relations between the tweets. Tweets are modeled as nodes in the network and an edge between two nodes is weighted to reflect how similar the two tweets are. To determine the similarity between tweets, our system leverages *hashtags*. Twitter defines hashtags used on their site in the following way.

"A hashtag—written with a # symbol—is used to index keywords or topics on Twitter. This function was created on Twitter, and allows people to easily follow topics they are interested in." [1]

Hashtags are essentially metadata allowing users to easily find tweets related to a certain topic. Moreover, hashtags many times indicate the stance that the user takes on the topic. A good example of this is the popular hashtag '#MAGA' which is an acronym for the campaign slogan claimed by the right wing incumbent president Donald Trump. A tweet with this hashtag can be identified as leaning right.

It may be argued that some hashtags are intentionally vague and simply used to boost a user's presence and may not actually provide information about the users position. A more optimistic interpretation of these hashtags is viewing them as a bridge between communities; a way for two disagreeing users to discover each others discussion circles and ideas. Additionally, labeling tweets based on their hashtags without context introduces some issues. For instance, a hashtag '#democrat' may be categorized as left leaning but in the tweet text it is actually being used to degrade the '#democrat' trend. Future work will be done to make our system more context aware.

From the sample of tweets related to a topic, the set of all unique hashtags is aggregated into a file and is manually labeled with a sentiment value existing in the set $\{-1, \frac{-1}{2}, 0, \frac{1}{2}, 1\}$ based on our best understanding of the trend being summarized by the hashtag. A hashtag with a sentiment score less than 0 is considered left leaning while a hashtag with a sentiment score greater than 0 is considered right leaning, as is illustrated in figure 2.

The un-directed weighted graph modelling the network can then be constructed using a data driven approach. Let $T = [t_1, ..., t_n]$ be the set of all the collected tweets for the topic being considered. Then, from the set of all hashtags existing in the sample of tweets, we can

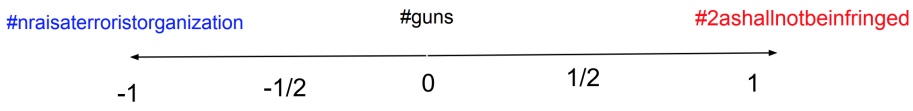


Figure 2: Polarity spectrum with hashtags from the topic 'guns'

build the $n \times m$ data matrix, call it $H = (h_{ij})$, such that

$$h_{ij} = \begin{cases} 1 \; if \; tweet_i \; contains \; hashtag \; j \\ 0 \; o.w. \end{cases}$$

Let $C = [c_1, ..., c_d]$, be a clustering assignment for the network, where each c_i in C is a list of nodes assigned to cluster i. Initially each node is assigned to its own individual cluster. Let $P = (p_{ij})$ be an $m \times d$ matrix such that entry p_{ij} is the proportion of occurrences of hashtag i contained in cluster j. Finally let $N = [n_1, ..., n_m]$ be such that n_i is the inverse of the number of occurrences of hashtag i. The network edges are then created with weights $w_{i,j}$ obtained using the following algorithm.

Algorithm 1 Update Network Edge Weights

for all t_i in T **do**
  $v_i \leftarrow \frac{h_{i,1} \cdot n_1 \cdot (p_{1,1},...,p_{1,d}) + ... + h_{i,m} \cdot n_m \cdot (p_{m,1},...,p_{m,d})}{h_{i,1} \cdot n_1 + ... + h_{i,m} \cdot n_m}$
end for
for all t_i in T **do**
  for all t_j in T **do**
    if $t_i = t_j$ **then**
      $w_{i,j} = 0$
    else
      $w_{i,j} = \phi(\frac{<v_i, v_j>}{||v_i|| \cdot ||v_j||})$
      $w_{j,i} = w_{i,j}$
    end if
  end for
end for
return $W = (w_{i,j})$

In this way, each tweet is assigned a d dimensional vector that describes their relationship to each of the of the d clusters. Then network edge weights, $w_{i,j}$, are calculated using the cosine of the angle between the vectors. The cosine similarity metric was chosen in this design since we are interested in specifically the direction that each vector is pointing in rather than the vector's magnitude. Additionally, in this case, since having a hashtag or not is simply a binary, 0 or 1, feature, this metric is equivalent to the Pearson correlation similarity [5].

If the cosine similarity was was purely used then, since the system only considers hashtags, there would be many disconnected components in the derived network model. Thus, to

introduce some probability for tweets to transition between communities that do not share hashtags, the cosine similarity metric is run through an exponential kernel ϕ. The parameters of the kernel can be chosen with the effect of changing the granularity of the clusterings.

As mentioned, initially each tweet is assigned to its own cluster. After the edges are created, the graph model is ran through the backward path clustering algorithm to obtain a new clustering assignment. The clustering obtained from the algorithm is then used to update the network edge weights using the same Update Network Edge Weights procedure. This process is re-iterated until convergence, that is until the same clustering is repeatedly returned from the backward path community detection algorithm. The entire network model building procedure is as is shown in the Build Network Model procedure shown in Algorithm 2.

Algorithm 2 Build Network Model

for all t_i in T **do**
  $c_i = [t_i]$
end for
repeat
  $c'_k \leftarrow c_k$ for all k
  $W = (w_{i,j}) \leftarrow$ Update Network Edge Weights
  $C = (c_k) \leftarrow$ Backward Path Community Detection
until $c'_k = c_k$ for all k

The advantage of this update procedure is that it is data driven and converges to a model which ensures that the clustering of the network remains fine grained enough to identify smaller communities in the network.

Community Detection: The community detection algorithm used in the system, backward path community detection, was developed at the University of Hawai'i Big Data Lab (Paravi and Santhanam, 2015)[2]. This particular algorithm captures an intuitive understanding of modern day information aggregation by modeling browsing activity with a slow mixing Markov process. The algorithm is probabilistic and leverages coupling from the past, a result which enables perfect sampling of the stationary distribution of a finite Markov chain [3].

The backward path community detection algorithm makes no assumptions on the number of communities present in a network and in fact provides potential clusterings at different granularities. Communities in the network are identified and returned by the algorithm in a hierarchical structure, starting with a fine grain clustering where all nodes are in their own community, and ending with the entire network identified as a single community. This feature of the algorithm is particularly advantageous to this application since the number of communities in our network is not known a priori and is in fact a characteristic we wish to uncover from the data.

However, the network update procedure and the polarity calculations require a single clustering. Thus, the hierarchical tree of clusterings provided by the backward path algorithm must be clipped at some level; a single clustering must be chosen from the retrieved set. This selection is made using the manually labeled hashtag sentiments. It is desirable to classify tweets with contrasting sentiments into different communities. Furthermore, a more coarse

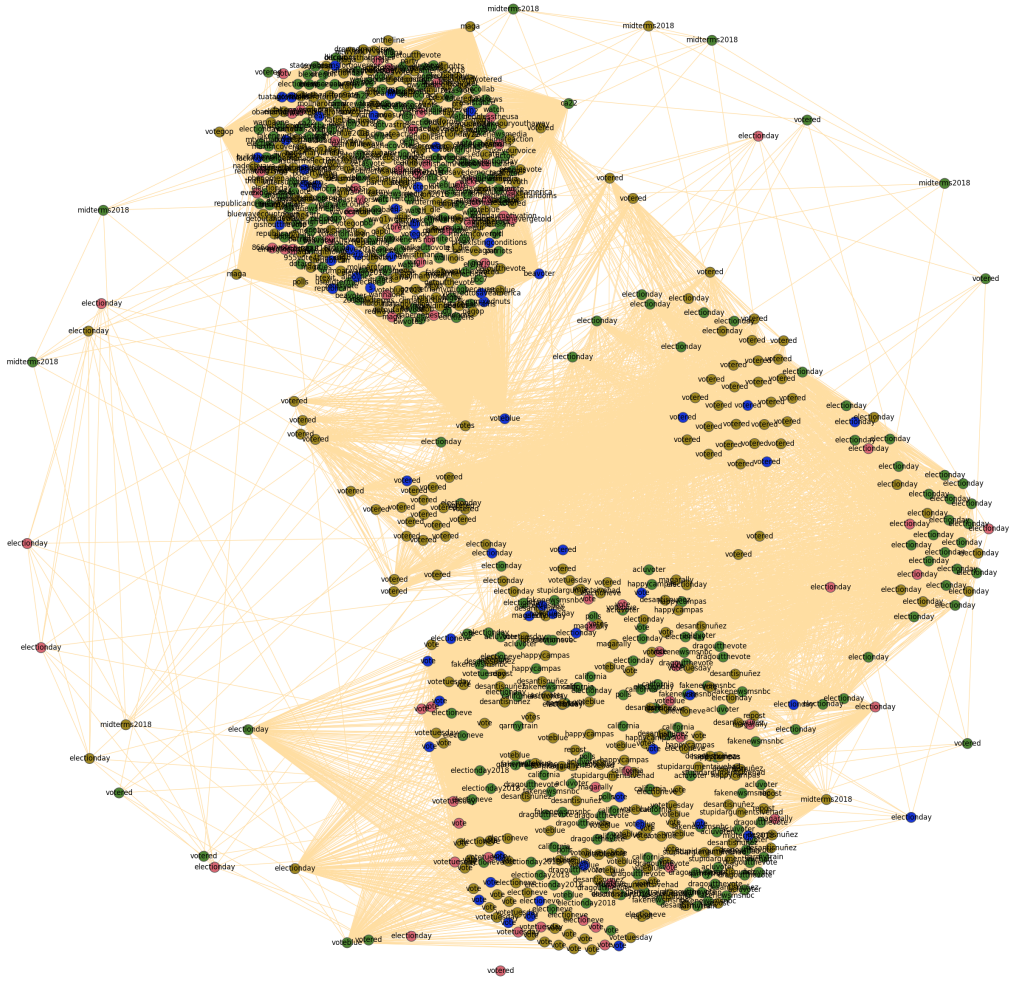
clustering, i.e. larger identified communities, provides more structure to the graph so that polarity calculations can be more informative and tractable. Thus, the clustering used to both update the edge weights of the graph is chosen to be the coarsest clustering which satisfies the constraint that no two tweets with contrasting average hashtag sentiment are found in the same community. If no such cluster is found, then the first level clustering of the backward path hierarchy is selected.

Final visualizations of the modeled network and the identified communities of this system can be seen in figure 3. These visualizations were made using the Python newtworkx and matplotlib libraries. The node layout is created using the spring layout which is an implementation of the Fruchterman-Reingold force-directed algorithm. Force directed algorithms attempt to layout the nodes so that edges are all of nearly equal length according to their edge weights. The Fruchterman-Reingold achieves this by simulating the motion of the nodes to minimize their energy [4]. The nodes of the network are tweets that are labeled with one of the hashtags presents in the tweet. Then the node is colored based on the community assignment.

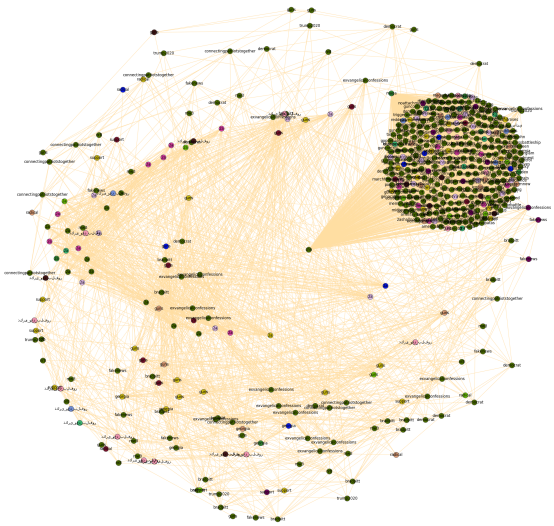
A sampling of tweet texts from identified communities of the 'immigration' data set is shown in table 3. These results show that not only are tweets with similar sentiment being clustered, but groups of tweets that are discussing and making similar points about a topic are identifiable. For instance, in table 3, tweets found in cluster 0 are all discussing housing issues related to immigration, tweets found in cluster 1 are using thanksgiving to make a point about the positive effects of immigration, while tweets found in cluster 2 are mostly discussing funding for construction of a new border wall.

Cluster	text
0	Lets get real on homelessness & ... #bbcqt
0	"1 in 200" are homeless,.. #bbcqt ... Blame endless immigration...
0	... taxpayer funded benefits and homes go to immigrants... bbcqt
1	#Thanksgiving.... Native-Americans understood... immigration are our strengths...
1	#Thanksgiving the commemoration of a day in which 50 Americans saved the lives of 52 illegal immigrants from England.
2	Mr. President we are some fed up Americans. We want to #BuildTheWall and be done with illegal immigrants.
2	#BuildTheWall Trump Super PAC Calls on Congress to Fund the Wall via BreitbartNews
2	PRESIDENT TRUMP... Border Wall Funding #BuildTheWall #CaravanInvasion #RWeBComingEurope
2	Deal with Mexico paves way for asylum overhaul at U.S. border @realDonaldTrump #BuildTheWall #TheGreatAwakening #QA

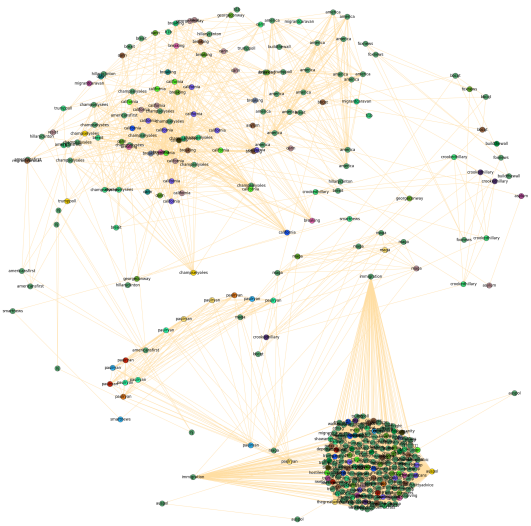
Table 3: Sampling of tweet texts from different communities identified from the 'immigration' data set



(a) Voting Rights and Turnout



(b) Gun Policies



(c) Immigration

Figure 3: Network model visualizations generated using the Fruchterman-Reingold force-directed algorithm for the three topics analyzed in this study

Polarity Calculations: Lastly, once the network clusterings have converged in the Build Network Model procedure, and the final community classification is made, the final task is to quantify the polarity. The polarity is measured at an individual community and network wide level for each topic.

The measure used to quantify the polarity between communities and the rest of the network is the cut conductance [6]. The conductance of a cut $(S, \overline{S})$ of a graph $G = (V, E)$ is defined as

$$\varphi(S) = \frac{\sum_{i \in S, j \in \bar{S}} w_{i_j}}{\min(w(S), w(\bar{S}))}$$

where $w_{i,j}$ is the weight of the edge from node i to node j and

$$w(S) = \sum_{i \in S} \sum_{j \in V} w_{i,j}$$

The conductance is a measure of how difficult it is to cross a cut during a random walk on the graph where the weights $w_{i,j}$ are normalized to reflect transition probabilities. A graph with a high conductance tells us that the cut is well-knit and not very polarized, while a low conductance tells us that the cut is highly polarized; the conductance and polarity are inversely proportional. This metric will be used at the individual community level to tell us how disconnected and polarized each community is in the network.

The polarity at the network wide level will be measured using an approximation of the graph conductance. The graph conductance is defined as the minimum conductance over all possible cuts.

$$\phi(G) = \min_{S \subseteq V} \varphi(S).$$

The approximation will be made by finding the minimum conductance of all possible cuts which disconnect the identified communities. Table 4 summarizes the results of the polarity calculations for each topic analyzed by the system.

Topic	Network Conductance	Number of Communities
Guns	0.0010697883742869512	14
Voting	0.00014492923268337674	3
Immigration	0.0007237128351882225	22

Table 4: Polarity Measurements using network conductance for each topic and the number of communities identified by clustering algorithm with 'Coarsest' clustering requirement discussed in the *Community Detection:* section.

The polarity calculations tells us that voting related tweets were the most polarized. The number of clusters also shows that more complicated topics such as immigration and gun control tend to have more schools of thought than something that is relatively more straight forward like voting rights.

Design Methodology

The implementation of the system follows an object oriented design methodology. The tasks are broken up and handled by five seperate classes: TweetCollector, TweetNetwork, TweetFeatureExtractor, Clusterer, and PolarityCalculator. Figure 4 illustrates the interaction between the classes and the separation of tasks. The benefit of such a design methodology is that it is tractable and can easily be extended for future development. The separation of tasks is designed so that changes in each module can be performed in parallel so long as the output meets the expected format. Furthermore, sharing instances of classes such as the TweetFeatureExtractor cut down on the need for repetitive computation and ensures certain attributes, such as the node id to tweet id map, are uniform across modules.

The TweetCollector class is responsible for setting up the connection to the Twitter search API, making search queries, and managing the storage of the collected data. The TweetFeatureExtractor class generates the necessary data frames required for downstream analysis in both the Clusterer and TweetNetwork class. The TweetNetwork class is responsible for driving the edge weight calculation procedure and storing the results. The Clusterer class wraps the backward path community detection algorithm and implements the procedure for obtaining the coarsest clustering satisfying the non-contrasting tweet sentiment constraint. Lastly, the PolarityCalculator class calculates the conductance of the final network model and clustering assignment obtained from the Clusterer class.

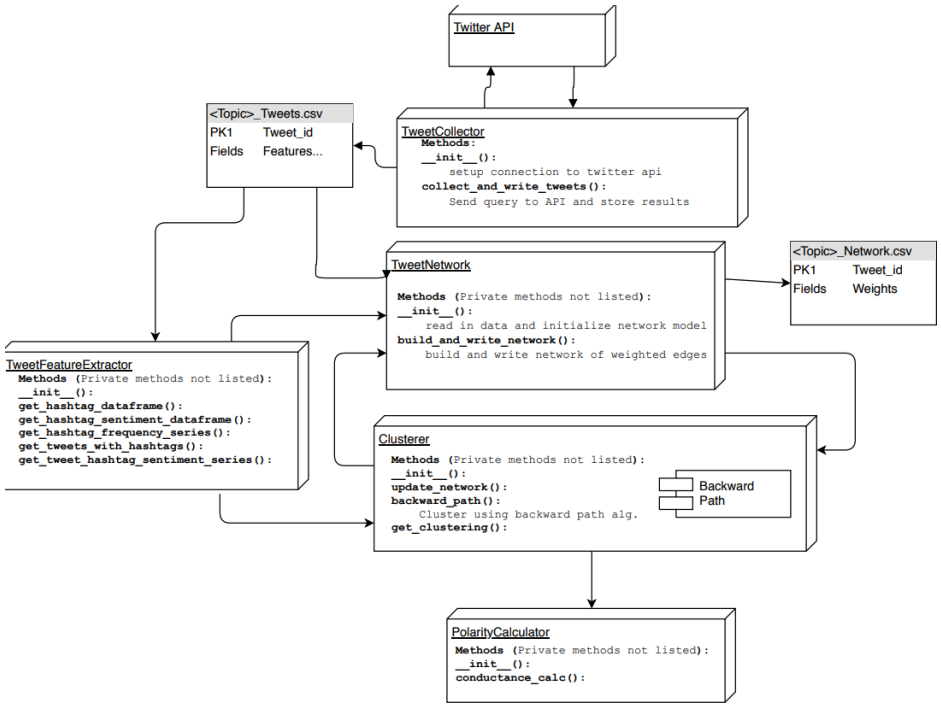


Figure 4: Class Structure for System

Related Work

Similar work has been done on this topic, most closely is that of Conover et al. (2011) [7]. This project differs primarily since it exclusively considered the left and right wings of the U.S. political spectrum and examined tweets during a specific time line, regardless of the topic being discussed. Furthermore, the network model developed by the researches is created based on whether two tweeters have either mentioned or retweeted one another. There are no edge weights is this model and cluster analysis is performed using label propagation. This advantage of this model is its simplicity, but it does not capture some of the complexities of the network such as the more tightly knit groups found within the larger communities.

Other work related to the polarization of information on the web includes that of Lai et al. (2015) [8], which analyzed the discourse of a single topic but was a soley Hashtag driven approach, ignoring other features which may be relevant. Moreover, the network created in this research modeled hashtags as nodes and connected these hashtags with the weight of edges determined on the number of tweets sharing the hahstags. Our approach models tweets as nodes and is intended to incorporate more features into the network to capture more complex relations.

Alternate Solutions

At different stages of the system, certain design decisions were made that could have alternative solutions. The alternate solutions may have varying consequences on the results.

As for retrieving relevant information and opinions, another approach to collecting tweets rather than using the Twitter search API would have been to use the perhaps one of Twitter's streaming data endpoints. The streaming data endpoints allow tracking tweets that are being posted in in current time rather than sampling from historical data. This approach would however restrict the data to a certain time frame rather than being able to sample data over the past month. Then some tasks such as analyzing how the polarity has changed over time is not as easily achieved.

To model the network, an alternative solution to the procedure implemented is to weight edges between tweets based on a simple heuristic such as the number of hashtags with the same sentiment shared between the tweets. This solution may however may not expose the finer subgroups of the network and only show the left and right ends of the spectrum.

The community detection algorithm used, backward path, was chosen due to its alignment with the intuitive understanding of the way users consume information on the web and the hierarchical set of results. Furthermore, the algorithm does not make an assumption on the number of clusters in the network, a critical feature of system.

Polarity is measured in the system using the conductance of the derived network model. This metric was chosen since the conductance can be thought of as a measure of the difficulty to break out of the most confined clusters of the network. This directly aligns with our interpretation of the polarity of information on the web. Another metric to possibly measure the polarity in the network is the modularity. The modularity of a network is the difference of the fraction of the edges that are within the identified communities and the expected fraction if the edges were distributed at random. This approach is interesting and may be implemented in future work and compared to the conductance measurements obtained.

Related Course Work

This research has been a demonstration of many of the skills I have accumulated from the engineering, computer science, and math coursework of my undergraduate studies. I have applied design strategies, problem solving skills, and project management strategies learned from EE260 and ICS314. The programming knowledge from EE160, ICS11, ICS211, and ICS212 gave me the background I needed to learn Python, the language used to implement the system. I have studied, designed, and implemented algorithms used in the systems which required understanding gained from ICS141, ICS241, and EE367. The theory and motivation of the process has come from coursework in linear algebra, probability and statistics, and machine learning, which I learned in the courses MATH311, MATH411, EE342, EE417, EE445, and EE491D.

Future Work

This implementation has been a proof of concept for the techniques used to build the network model, identify communities, and measure polarity. The system will be developed in future work to consider additional features such as tweet popularity, tweet text sentiment, and more. Moreover, the workflow for collecting the tweets can be modified to minimize bias introduced by using human intuition to create search queries.

One way to develop search queries to minimize bias would be to seed the search with an intuitive set of phrases that is felt to capture most of the existing ideologies, then, using the results of the search, the system can dynamically create a new set of phrases by using the set of *hashtags* of the retrieved tweets for another query. This methodology could be scaled to identify relevant sources of information generically by using state of the art auto tagging algorithms to extract keywords and phrases from the sources. Collection can start with a Google search for a topic using a search string that is as neutral as possible. Then a first pass scraping and analysis can be run. The most common keywords and phrases from different groups identified in the first pass can be used to dynamically form a new query.

Engineering Standards and Practical Constraints

- Economic

  This project is closely related to recommender systems and sentiment analysis. Reccomender systems are used by many online services to predict what users would like and dislike and have had a serious impact on the economy in recent years by optimizing marketing strategies for businesses. The goal of sentiment analysis is to understand how users feels about certain topics. Sentiment analysis can be used as a feature to predict the trend of a stock or the success of a new movie.

- Environmental and Sustainability

  The environmental and sustainability impacts of this research are minimal. There is, however, potential in using the system to better understand the communities discussing the topics. This could lead to more productive ways to target and educate misinformed individuals.

- Manufacturability

  This system has proven to be manufacturable from the analysis and results reported. The computing tasks for this report were all performed on a personal computer and completed within a reasonable run time.

- Ethical

  The ethical concerns with this system include the potential to infer a users' political beliefs. Some users may not wish for this type of information to be known. Furthermore, there is a potential for malicious persons to misuse this information, a prime example of this is Russian meddling in the 2016 U.S. elections [9]. To ensure that this type of misuse of data does not occur Twitter enforces a strict policy that no data collected from the API is to be made available online. Furthermore, before granting access to the search API Twitter does a screening to ensure the application you are creating does not break their code of ethics. The analysis in this report does not share information about any particular user but rather the network as a whole.

- Health and Safety

  Since no data is shared about any particular user in this study and only information about the entire network as whole is being analyzed there is little effect on an individuals health and safety. This research is intended to support healthy discourse of difficult topics. By identifying highly polarized communities, we can build bridges of communication and recognizes and appreciate other schools of thought

- Social

  An objective of this study is to better understand the state of discourse at a topic specific level. If the polarity of discussions can be quantified and the primary sources of the disagreement identified, then solutions to the issue can be made.

- Political

  An understanding of the polarization of information on the web could be used to help policy makers identify topics of controversy or agreement to better serve their constituents. Furthermore, campaign strategies could greatly benefit from information about how different communities feel about topics.

Acknowledgements
Special thanks to CSoI: Center for Science of Information (https://www.soihub.org) is a National Science Foundation Science and Technology Center made possible under grant NSF CCF- 0939370

References

[1] Help.twitter.com. (2018). How to use hashtags. [online] Available at: https://help.twitter.com/en/using-twitter/how-to-use-hashtags [Accessed 26 Aug. 2018].

[2] Torghabeh, Ramezan Paravi, and Narayana Prasad Santhanam. "Community Detection Using Slow Mixing Markov Models." 2015, pp. 1-11.

[3] J. G. Propp and D. B. Wilson, "Exact sampling with coupled markov chains and applications to statistical mechanics," Random structures and Algorithms, vol. 9, no. 1-2, pp. 223–252, 1996.

[4] Thomas M. J. Fruchterman and Edward M. Reingold, "Graph Drawing by Force-directed Placement," Software-Practice and Experience, vol. 21, no. 1-1, pp. 1129-1164, 1991.

[5] L. A. Hassanieh, C. A. Jaoudeh, J. B. Abdo and J. Demerjian, "Similarity measures for collaborative filtering recommender systems." 2018 IEEE Middle East and North Africa Communications Conference (MENACOMM), Jounieh, 2018, pp. 1-5.

[6] A. Sinclair, M. R. Jerrum, "Approximate counting uniform generation and rapidly mixing markov chains", Information and Computing.

[7] M. Conover, J. Ratkiewicz, M. Francisco, B. Goncalves, A. Flammini, and F. Menczer, "Political Polarization on Twitter." 2011 Fifth International AAAI Conference on Weblogs and Social Media, Barcelona, 2011, pp. 1-8.

[8] M. Lai, C. Bosco, V. Patti and D. Virone, "Debate on political reforms in Twitter: A hashtag-driven analysis of political polarization." 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Paris, 2015, pp. 1-9.

[9] Office of the Director of National Intelligence (2017). Background to "Assessing Russian Activities and Intentions in Recent US Elections": The Analytic Process and Cyber Incident Attribution. [online] Available at: https://www.dni.gov/files/documents/$ICA_2017_01.pdf[Accessed1Dec.2018]$.