Community Detection and Trend Analysis of #MeToo Event on Twitter

Bhavika Reddy Jalli
bhavikaj@umich.edu

Cixing Li
cixingli@umich.edu

Siyan Chen
siyanc@umich.edu

Vidya Mansur
vidyam@umich.edu

ABSTRACT

The original purpose of #MeToo by creator Tarana Burke was to empower women through empathy, especially the experiences of young and vulnerable brown and black women. In October 2017, hashtag #MeToo was used extensively on social media especially on Twitter to help demonstrate the widespread prevalence of sexual assault and harassment, especially in the workplace. The hashtag has now spread across more than 85 countries. In this project, we seek to detect communities that are formed based on implicit and explicit relationships through social interactions. To this end, we create a Re-tweet network, a Mention network and a Similarity based network with the tweets containing the #Metoo tags and detect meaningful communities using Label Propagation Algorithm, Louvain Modularity and InfoMap community detection algorithms. To further understand the communities and their structure, we present a temporal analysis of the trends in these communities and then compare the performances of these community detection algorithms using goodness metrics.

KEYWORDS

Twitter, #MeToo, Community Detection, Label Propagation Algorithm, Louvain Modularity, InfoMap, temporal analysis, Graph Visualization

1 INTRODUCTION

#MeToo event is a widely popular event on Twitter. The #MeToo event spread virally in the months of October 17^{th} and November 17^{th} after it was popularized by actress Alyssa Milano. It is seen that limited work has been done in analyzing the trends of this event on Twitter using graph techniques. The dataset for the project was prepared by crawling the tweets with #MeToo tag during the time period 25^{th} February'18 to 31^{st} March'18. We detect communities in the #MeToo event with dense within-group connections and meager between-groups connections based on Re-tweet network, Mention network and the Similarity based graph using Label Propagation algorithm(LPA), Louvain Modularity(LM) and InfoMap algorithms. We have chosen these algorithms as we want to find distinct communities in our networks and also because LM and InfoMap can both be used for weighted graphs. These three algorithms are also efficient in their run time and can be scaled to large graphs.

With the identification of the communities, we present the insights on the polarization present in the conversations and also the analysis of the users/tweets in the communities. With the detection of the communities, we identify the implicit connections that go beyond the friends/followers circle. These communities contain users with similar interests and thoughts. Hence, messages and ads can be structured according to their characteristics. We discerned the temporal evolution of the sentiments in the detected communities to show the human reactions to such movements. We also present the geographical distribution of these tweets to get the sense of how widely spread is this event. We then establish a comparison of the performance of the community detection algorithms employed using goodness metrics for e.g conductance and modularity.

{"replied_to": null,
"text": "RT @jfkeeler: My experience of setting up the blog and advice for others wanting to do the same :) #Metoo https://t.co/oNhT7R6tQy",
"hashtags_in_the_tweet": ["Metoo"],
"created_at": "2018-02-26 19:55",
"name": "Rain Clan Dakh\u00f3ta",
"screen_name": "Mdewakanton",
"retweet_count": 7,
"location": "Tinta Winta, Turtle \ud83d\udc22 Island",
"user_ID": 175514174,
"Mentions_in_the_tweet": ["jfkeeler"]}

Figure 1: The Raw Tweet data collected using Tweepy

2 DATA

2.1 Data Collection

In this project we use the Twitter API, Tweepy to collect the data from 25^{th} February'18 to 31^{st} March'18. Tweepy is open-sourced, hosted on GitHub and enables Python to communicate with Twitter platform through its API. We created a Twitter application to instantiate the Tweepy API. Twitter data access can be achieved through two ad hoc APIs that represent different Twitter features: Stream and Representational State Transfer (REST). The 'Stream API' is focused in data mining providing the real time sample of the tweets. The 'REST API' enables developers to access some of the core primitives of Twitter including time-lines, status updates, and user information. We extracted over 784,000 tweets from the duration stated above using the 'REST API' with the query search = "#MeToo".

Raw Text : "RT @jfkeeler: My experience of setting up the blog and advice for others wanting to do the same :) #Metoo https://t.co/oNhT7R6tQy"
Processed Text: experience setting blog advice others want

Figure 2: A look at the same tweet text before and after processing

The raw tweet obtained is stored in the form of a python dictionary and eventually a list of the tweets is created. Each of the raw tweet requires pre-processing. We extract the fields screen name, re-tweet_count, replied_to, text and location.

The screen name, re-tweet count, replied-to fields can be extracted easily from the raw tweet where as the text, location field require pre-processing which is explained in the below section.

2.2 Data Pre-Processing

There were two fields that required focused pre-processing. These were the tweet and the location.

2.2.1 Tweet. Extracting the re-tweeted person and the original owner of the tweet are very important to form the re-tweet network. The re-tweets always start with the letters 'RT'. Using this fact, we were able to extract the user name of the original tweet's owner. The same can be done with the mentions in the tweet text. The pre-processing of the tweet text consisted of removing the URLs, hashtags, mentions,smilies and other punctuation. We used python libraries and regular expressions to remove the unwanted characters using the commands re.sub and replace. The mentions and hashtags were removed as the they have a very indicative # or at the beginning of the user names.

Once the tweet was broken down to only English words, we then used Natural language toolkit (NLTK) library to remove the stop words such as *a, an, the, and* etc. The words were then lemmatized using the WordNetLemmatizer. The WordNetLemmatizer does full morphological analysis to accurately identify the lemma for each word. This helps in word to vector conversion for sentiment analysis or cosine similarity measurement.

2.2.2 Location. From the Tweepy API we were able to extract the location and place fields which are not present for all the tweets. The place field gives the tweet_city, tweet_state and tweet_country as separate entities and location field gives a string of either (city,state) or (state,country). We came up with an uniform way to represent the location for each tweet. This also includes some pre-processing such as removal of emojis.

We intend to analyze the tweets geo-graphically. With this in mind we tried to use geocode package which allows to extract the geo-graphical location (latitude and longitude) of the tweet.

3 PROPOSED METHOD

3.1 Top Level Architecture

After collecting and pre-processing the tweets, we create three networks Mention network, Re-tweet network, Similarity Based network described in detail in the next section. The figure 3 shows

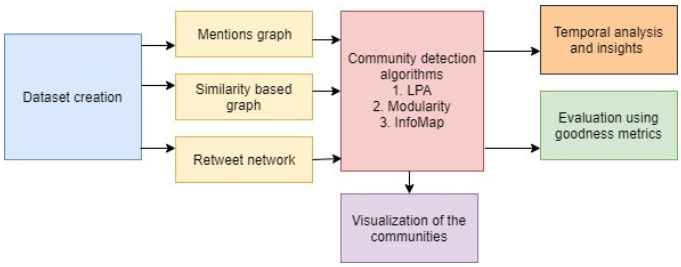


Figure 3: Work Flow of the Project

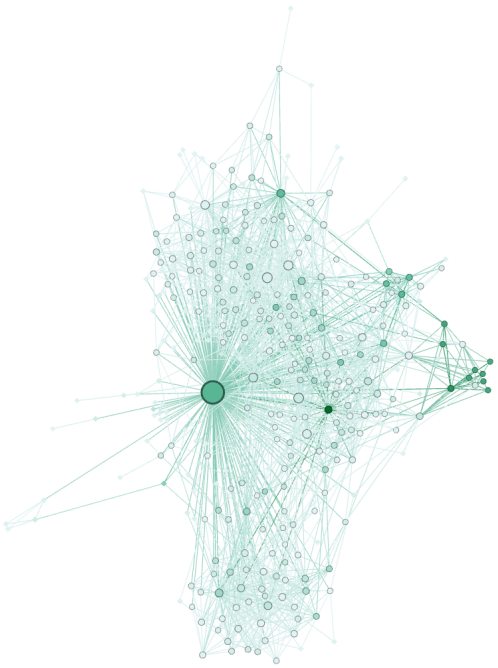


Figure 4: The 500 biggest nodes in the Mention network where the users are connected based on their mentions

the work flow of our project. All of these networks will be clustered using three community detection algorithm-Label Propagation Algorithm, Louvain Modularity and InfoMap. The communities thus obtained will be evaluated using Modularity and Conductance metrics. The communities will also be visualized and the top nodes in each community will be looked into to gain insights about the community. We will also divide the Mention graph into three time periods and provide an analysis of how the communities evolve over time.

3.2 Network Creation

3.2.1 Mention Network. After pre-processing the data we first create a Mention network $G_M(V, E)$. The nodes (V) of the graphs are the users and the edges (E) are established by connecting the

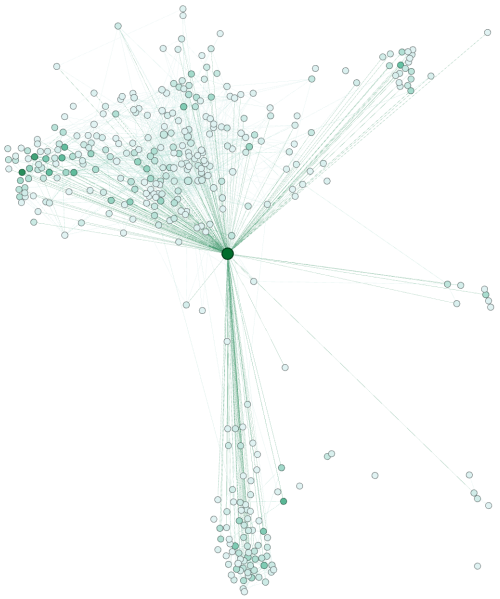


Figure 5: The Re-tweet network where the users are connected based on the re-tweets

Topic #0: kiss muslim american mental leaving idol giving turned lose health

Topic #1: ha first change handle movement state since miller backlash police

Topic #2: jennifer lopez face story saw act terrified start shares outrage

Topic #3: wa would come day may end child tweet could case

Topic #4: people guy victim movement not ha weinstein harvey help got

Topic #5: woman sex right know want submissive happy single party incredible

Figure 6: 5 topics extracted from LDA

user and all the other users that were mentioned in the tweet. We then used this network to find meaningful communities. The intuition here is that people supporting what the others have said tend to mention them in their tweets and hence belong to the same community. Please refer to the Figure 4 for the Mention network with the most mentioned users having the bigger nodes. The Mention network has 354,853 nodes and 624,028 edges.

3.2.2 Re-tweet Network. In a Re-tweet network G_R(V,E), the nodes are the users (V) and the edge (E) connect the user and the re-tweeted user. This Re-tweet network has 294,230 nodes and 395,340 edges. Refer to the Figure 5 for the Re-tweet network, where size of the node corresponds to number of times the user has been re-tweeted. It is inferred that the communities detected in the the Re-tweet network represents much stronger connection as re-tweeting a user would mean direct agreement or disagreement to the content. The communities formed here will be based on direct relationships.

3.2.3 Similarity Based Network. To create a Similarity based network, we first represent the tweets into a vector-space model. The method we use is called Latent Dirichlet Allocation(LDA)[2] which is a generative probabilistic model. We can view LDA as a dimensionality reduction technique. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.

LDA assumes the following generative process for each document w in a corpus D:

(1) Choose $N \sim Poisson(\xi)$
(2) Choose $\theta \sim Dir(\alpha)$
(3) For each of the N words w_n:
    (a) Choose a topic $z_n \sim Multinomial(\theta)$.
    (b) Choose a word w_n from $p(w_n|z_n, \beta)$,a multinomial probability conditioned on the topic z_n.

The dimensionality k of the Dirichlet distribution(and thus the dimensionality of the topic variable z) is assumed known and fixed. The word probabilities are parameterized by a $k \times V$ matrix β where $\beta_{ij} = p(w^j = 1|z^i = 1)$ which we treat as fixed quantity that is to be estimated. A k-dimensional Dirichlet random variable θ can take values in the (k-1)-simplex and has the following probability density on this simplex:

$$p(\theta|\alpha) = \frac{\Gamma(\Sigma_{i=1}^{k}\alpha_i)}{\Pi_i^k \Gamma(\alpha_i)}\theta_1^{\alpha_1-1}\cdots\theta_k^{\alpha_k-1} \tag{1}$$

where the parameter α is a k-vector with components $\alpha_i > 0$. In order to use LDA, the key inferential problem that we need to solve is that of computing the posterior distribution of the hidden variables given a document(which is the feature vector of a tweet in this case):

$$p(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)} \tag{2}$$

LDA uses Variational inference method to estimate this distribution.

After pre-processing, we generate a tweet-term matrix A of order $m \times n$ where m is the number of tweets and n is the number of tokenized terms in the corpus. The elements in matrix A, a_{ij}, is determined as the frequency of a term t_i in j^{th} tweet. We fit out LDA model on matrix A and generate a $m \times k$ feature matrix Φ where k is a hyper-parameter corresponding to the number of latent topics. The j^{th} row of Φ is the distribution of j^{th} tweet on k latent topics and also the feature vector of this tweet.

The Similarity based graph is generated as a weighted graph in which tweets represent nodes and similarity value between a pair of tweets is considered as a weighted edge between them. We use cosine similarity to calculate the similarity between a pair of tweets. The cosine similarity of two feature vectors a and b can be calculated using following equation:

$$Cosine(a, b) = \frac{a \cdot b}{||a|| \cdot ||b||} \tag{3}$$

The Figure 7 shows the Graph based on cosine similarity. It contains 14,228 nodes and 1,025,793 edges.

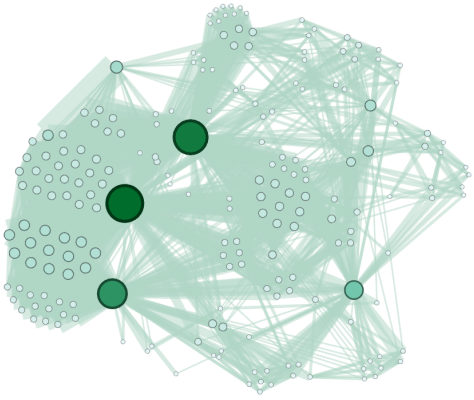


Figure 7: The Similarity based graph

3.3 Clustering Methodology

3.3.1 LPA. For the Mention network, we propose Label Propagation Algorithm (LPA) as it could cluster nodes through an unsupervised-learning approach. LPA has a simple and clear structure. As the first step we initialize every node to a unique label.

At the first step we randomly choose a node and change its label to the label of the community to which most of its neighbors belong to. If multiple labels satisfy this condition, we will choose the label of the node with highest total degree. The reason to make this choice instead of choosing it randomly is to speed up the clustering procedure as well as to obtain consistent results. If there are still multiple labels we can choose, we will simply pick one label randomly.

Once we change the label of a node, we will update it immediately which is referred to as asynchronous update method. Asynchronous update method avoids oscillation problem which could happen in synchronous update method. For example, if one node is surrounded by several nodes from another community, it will change it's label to the label of it's neighbors and all it's neighbors will change their label to the original label of this 'hub' node. And this procedure will not end and we call such a situation as oscillation.

We will repeat this step of updating the labels until every node shares a same label with most of its neighbors, i.e. no nodes will change its label. The Figure 8 represents the flowchart of the LPA algorithm.

3.3.2 Louvain Modularity. We propose Louvain Modularity [1] as one of the algorithm to extract communities. Modularity, which measures the density of links inside communities to links outside communities, is a value in (-1, 1) and defined as:

$$Q = \frac{1}{2m}\Sigma_{ij}[A_{ij} - \frac{k_i k_j}{2m}]\delta(c_i, c_j), \tag{4}$$

where A_{ij} is edge weight between node i and j, k_i and k_j are sum of edge weights connected to node i and j respectively, m is sum of

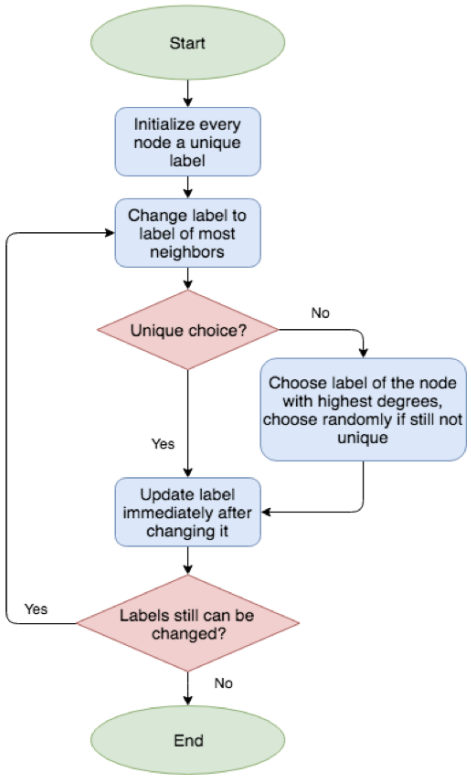


Figure 8: LPA Flowchart

weights of all edges in the graph, δ is a simple delta function and c_i, c_j are node i and node j respectively.

Each node initially is assigned to its own community and we calculate the delta modularity if we move a node i to the community of its neighbor node:

$$\Delta Q = [\frac{\Sigma_{in} + k_{i,in}}{2m} - (\frac{\Sigma_{tot} + k_i}{2m})^2] - [\frac{\Sigma_{in}}{2m} - (\frac{\Sigma_{tot}}{2m})^2 - (\frac{k_i}{2m})^2], \tag{5}$$

where Σ_{in} is sum of edge weights in the new community, Σ_{tot} is sum of weight of all edges linked to nodes in this new community and $k_{i,in}$ is sum of edge weights in this new community.

To maximize the modularity, we confirm the movement if maximum ΔQ is larger than 0. This process is applied repeatedly and sequentially and will end until no modularity increase occurs. Then the second step is to group nodes in the same community into one bigger node, all links between nodes in two communities is defined as a weighted edge between two communities. Thus, we can achieve a new graph and apply the first step iteratively. The Figure 9 represents the flowchart of the Louvain Modularity algorithm.

3.3.3 InfoMap. InfoMap uses the probability flow of random walks on a network as a proxy for information flows in the real

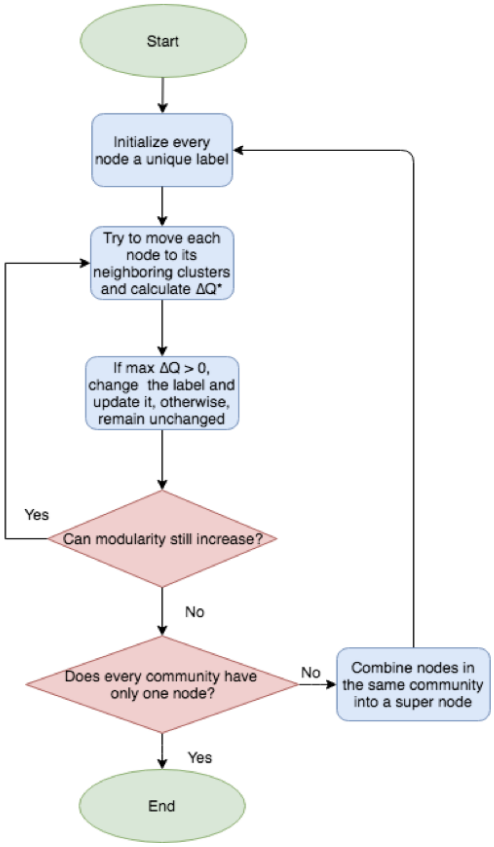


Figure 9: Louvian Modularity Flowchart

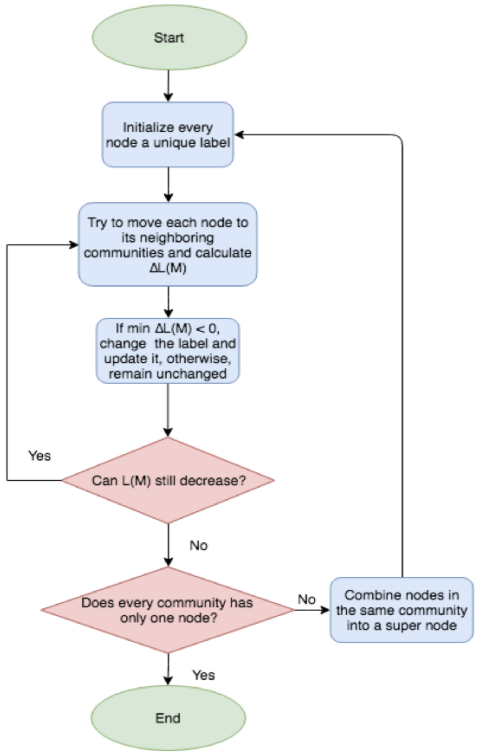


Figure 10: InfoMap Flowchart

system and decompose the network into modules by compressing a description of the probability flow. Taking this approach, InfoMap uses an efficient code to describe a random walk on a network and thus finding community structure in networks is equivalent to solving a coding problem.

We aim to choose a code that will allow us to efficiently describe paths on the networks that arise from a random walk process in a language that reflects the underlying structure of the network. Many real- world networks are structured into a set of regions such that once the random walker enter a region, it tends to stay there for a long time, and movements between regions are relatively rare. We can take a region with a long persistence time and give it its own separate codebook. So long as we are content to reuse codewords in other regional codebooks, the codewords used to name the locations in any single region will be shorter than those in the global code, because there are fewer locations to be specified. Using multiple codebooks, we transform the problem of minimizing the description length of places traced by a path into the problem of how we should best partition the network with respect to flow. We do not need to devise an optimal code for a given partition to estimate how efficient that optimal code would be. The main idea is how efficient the optimal code would be for any given partition, without actually devising the code. To find an optimal partition of the network, it is sufficient to calculate the theoretical limit for different partitions of the network and pick the one gives the shortest description length.

Here comes the map equation. For a module partition M of n nodes $\alpha = 1, 2, ..., n$ into m modules, the lower bound on code length is defined as $L(M)$. According to Shannon's source coding theorem, when you use n codewords to describe the n state of a random variable X that occur with frequencies p_i, the average length of a codeword can be no less than the entropy of the random variable X itself: $H(X) = -\Sigma_1^n p_i log(p_i)$(we measure code lengths in bits and take the logarithm in base 2). This provides us with a lower bound on the average length of codewords for each codebook.The map equation is :

$$L(M) = q_{\curvearrowright} H(Q) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(P^i) \tag{6}$$

Here $H(Q)$ is the frequency-weighted average length of codewords of module index and$H(P^i)$ is frequency-weighted average length of codewords in module codebook i. Further, the entropy terms are weighted by the rate at which the codebooks are used. With $q_{i\curvearrowright}$ for the probability to exit module i, the probability that the random walker switches modules on any given step $q_{\curvearrowright} =$

$\Sigma_{i=1}^{m} q_{i\curvearrowright}$. With p_α for the probability to visit node α, module codebook i is used at a rate $p^i_{\circlearrowleft} = \Sigma_{\alpha \in i} p_\alpha + q_{i\curvearrowright}$,the fraction of time the random walk spends in module i plus the probability that it exits the module

As Figure 10 shows, the core of the algorithm follows closely the Louvain method, the difference is the goal of InfoMap is to minimize the map equation rather than modularity.

4 EVALUATION PLAN

The data collected by the team does not have any information about the ground-truth communities. Due to this reason, we have decided to use goodness metrics to quantitatively measure structural properties of the identified communities instead of evaluation metrics such as F1-score, Precision and Recall. The paper [3] helped us choose the metrics Modularity and Conductance.

Given an undirected graph $G(V, E)$ and a set of nodes $S \subseteq V$, let $E_s = |\{(u, v) \in E | u, v \in S\}|$ be the number of edges in the subgraph induced by S. Let $O_s = |\{(u, v) | u \in S, v \notin S\}|$ be the number of edges between the vertices in S and any vertex outside of S. Given these notations, the goodness metrics for a community S are defined as follows:

(1) Modularity: the fraction of the edges that fall within the given groups minus the expected fraction if edges were distributed at random. Therefore a higher modularity indicates a cluster with dense connection within the cluster and sparse connections with nodes outside the cluster. The equation for modularity is given by

$$Q = \sum_{i=1}^{c} (e_{ii} - a_i^2)$$

where e_{ij} is the fraction of edges with one end of the vertex in community i and the other in community j and a_i is the fraction of ends of edges that are attached to vertex in community i and $a_i = \sum_j (e_{ij})$

(2) Conductance $\frac{O_s}{2E_s + O_s}$: the fraction of edges that point outside the community. This captures the surface area to the volume of the community. Therefore a lower conductance value indicates a denser community.

5 EXPERIMENTS

5.1 Geographical Distribution of the tweets with #MeToo

We have selected 1000 tweets randomly from the 784,000 tweets and obtained their location using the geocode package as explained in the data pre-processing step. Using the latitude and longitude of the obtained location, and the basemap python package, we plotted the tweets on a map. From the Figure 11, we can see that the dialogue on #MeToo consisted people from all over the world. There is a higher concentration in North America and Western Europe. This could be because of two reasons. One reason is that there is a larger focus on the event in these areas. Secondly, only tweets in the English language were selected and hence a natural concentration in the English speaking areas around the globe.

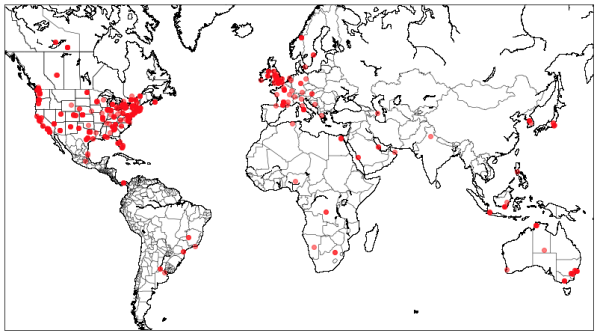


Figure 11: Geographical distribution of tweets with #MeToo

5.2 Community Detection of the Mentions Network

5.2.1 Using Label Propagation Algorithm. The LPA has successfully converged and found 11023 communities in the Mention graph. The communities have been visualized using the Gephi Graph Visualization tool. The Figure 12 shows the mention graph before and after clustering using the Label Propagation Algorithm.

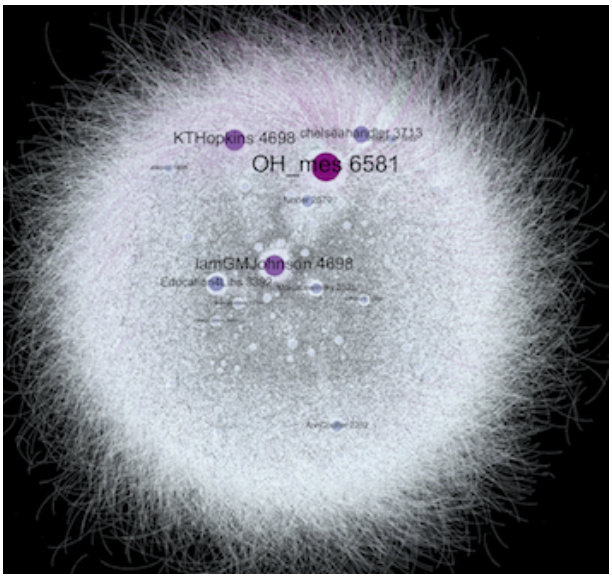
5.2.2 Using Louvain Modularity. One of the main goals of this project is to detect communities that go beyond the explicit relationships of followers and friends. We want to find communities that are formed based on similar interests and opinions due to social interactions. A key idea here is to use the re-tweets and mentions as a way people agree with others ideas. A person mentioning two users in their tweets will help form an edge between these users who might have never interacted but have now been connected through an implicit relationship. To create a sparse Mention network, we deleted the nodes with degree = 1.

We have discovered three such communities based on implicit relationships. For simplicity let's call them the Green, Red and the Blue communities. The table 1 gives a brief look into the characteristic of these communities.

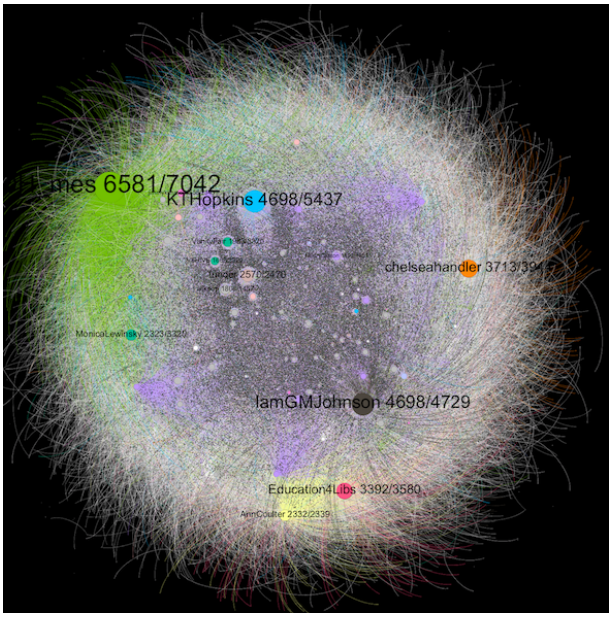
Community	Analysis
Green	Consists of News outlets, journalists, authors and columnists
Red	Consists of users with conservative views, politically right leaning with dismissive views towards the #MeToo event
Blue	Consists of women specific organizations, activists and liberal thinkers with extreme support

Table 1: Characteristics of the top three communities in the mentions graph

The Green community consists of news outlets, journalists, writers, and columnists. The members post neutral tweets about the event that are mostly informational content. The Red community consists of users with conservative views who are politically right leaning with tweets indicating a dismissive view towards the



(a) The Mention graph before LPA with the total degree of each node



(b) The Mention graph after LPA with 11023 communities with the total degree and the total number of nodes in the community

Figure 12: Community Detection using Label Propagation Algorithm

#MeToo event. Finally, the Blue community consists of women specific organizations, activists and liberal thinkers with extreme support to the event.

We have divided the entire mentions graph into three time periods to see the evolution of these communities.

(1) February 25^{th} to March 9^{th}
(2) March 10^{th} to March 19^{th}
(3) March 20^{th} to March 31^{st}

Figure 13 shows the evolution of the communities in the mentions graph. The communities are dynamic in their membership. That is the nodes in the graph aren't the same during the entire duration. However, the communities still retain their characteristics. For example, the Green community in the period February 25^{th} to March 9^{th} consists of Washington Post which is a news outlet and the Green community in the period March 20^{th} to March 31^{st} does not have Washington Post but instead consists of CNN which is also a news outlet.

5.3 Community Detection in Re-tweet Network

In the Re-tweet network the users are connected if either has re-tweeted the other and the frequency of the re-tweets is the weight of the edge. To create a sparse Re-tweet network, we deleted the nodes with degree = 1. The new Re-tweet network contains 73,149 nodes and 181,577 edges. We implemented the LPA, LM and InfoMap community detection algorithm on the re-tweet graph. The number of communities detected using LPA are 1582 and 549 communities using LM. The figure 14 shows the top 3 communities obtained by applying Louvain Modularity to the Re-tweet network.

The relationships in the Re-tweet network imply a direct connection between the users and that the membership in each community is exclusive. The members in the community display a strong binding. Table 2 gives a brief look into the characteristics of these communities.

Community	Analysis
Pink	Consists of Korean entertainment news page and mainly people from Korea re-tweeting
Purple	Consists of people with very strong political views.
Mustard yellow	Consists of people who are heads of famous organizations e.g Jon Cooper - Chair, Democratic Coalition, Amy Siskind - President, New Agenda and extend support to the movement

Table 2: Characteristics of the top three communities in the Re-tweet network

5.4 Latent Dirichlet Allocation and Cosine Similarity

LDA has been proven to work well on tweets[4]. We also tried other feature extraction methods like Non-negative Matrix Factorization(NMF). However, this method which is based on Term Frequency-Inverse Document Frequency(TF-IDF) is not suitable for short documents like tweets. Besides, the coherent topics identified by LDA is more desirable in calculating cosine similarity than the incoherent topics given by NMF. However, our method still has some limitations. There is no good ways of evaluating the topics generated by LDA and this requires us to look at them and see if they make sense. It is also highly time consuming to calculate pairwise cosine similarity for a large network.

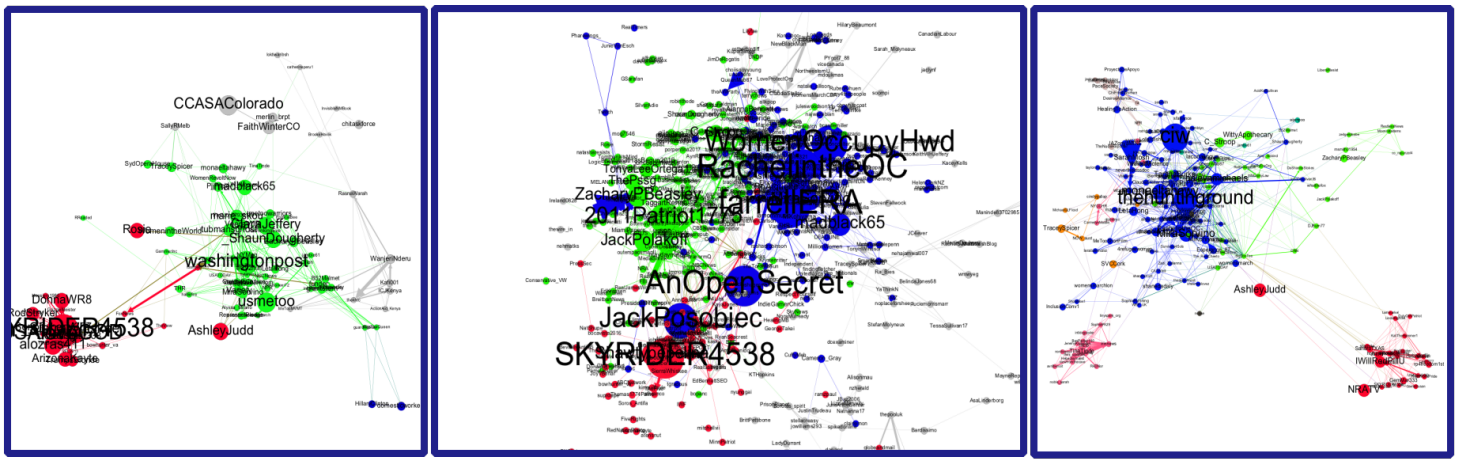


Fig.A Communities for the period Feb 25-Mar 9 Fig.B Communities for the period Mar 10-Mar 19 Fig.C Communities for the period Mar 20-Mar 31

Figure 13: Evolution of the communities in the mention graph over three time periods

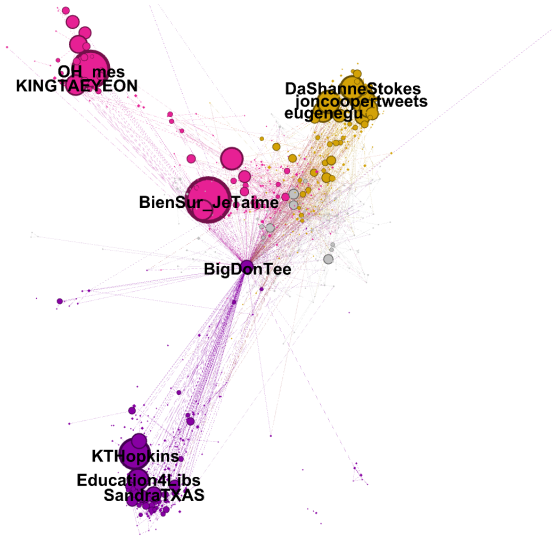


Figure 14: Communities in the Re-tweet network

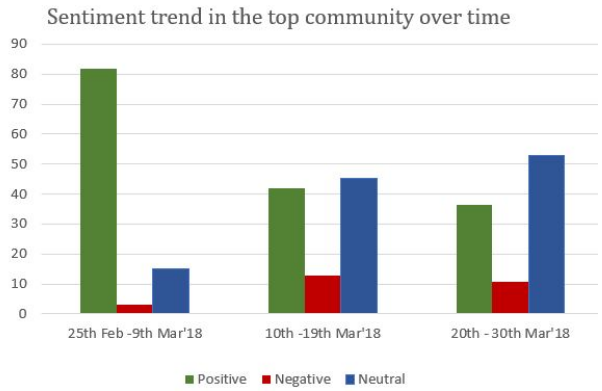


Figure 15: Sentiment Analysis Tweets e.g

We randomly sample 14,228 tweets to generate a Similarity based network of our dataset following the above method. In our implementation, we set the number of latent topics k to 30 and α, β to 0.01. After feature extraction, we compute the pairwise similarity between each two nodes, and pick weights > 0.7 as valid edges. The number of edges in this network is 1,025,793.

5.5 Sentiment Analysis

In this section, we present how the sentiment of the people changes over a period of time towards the #MeToo event. We applied TextBlob [a library for processing textual data and perform Natural Language Processing (NLP) tasks], to extract the sentiment of the tweets present in the dataset. In this regard, we created a Similarity based network with the tweets as the nodes with edges connecting the nodes if the cosine similarity value between the tweets is greater than the threshold. The similarity value is then the weight of the edge. We ran the InfoMap community detection algorithm on this and found distinct communities. Figure 17 represents the communities detected on the similarity graph. The table shows the sentiment distribution in the top 3 communities. The Figure 16 shows the top words (most frequently used) in the top community, which gives us a sense of the tweets.

Communities	Number of nodes	Prominent Sentiment
Green	1086	Positive 74%
Blue	1073	Neutral 53%
Red	876	Positive 48%

Table 3: Sentiment Analysis

Figure 16: Most frequent words used in the top community

Figure 17: Sentiment Analysis Tweets e.g

Intuitively we see that the number of people in support of the #MeToo event are larger compared to the percentage of people who are not supportive. The figure 15 presents the sentiment trend over time in the top community. We see a sharp spike in the positive comments during the 1st time interval as it was the week of International women's day.

5.6 Evaluation of the Methodologies

Table 4 gives the Modularity and the Conductance score of the Mention network, Re-tweet network and the Similarity based network. From the table we can see that the Louvain Modularity method attains the best modularity for the Mention network and the Re-tweet network where as the InfoMap method attains the least conductance. InfoMap yields the highest modularity and the least conductance for the Similarity based graph indicating that the community detection by InfoMap for this graph attains the best results.

By comparing the three algorithms on community detection, we came to the following conclusions. The LPA algorithm works the best on graphs with very well defined communities. A well defined community is a community with dense connections within the community and sparse connections to the nodes outside the community. Hence, LPA has the highest modularity score on the Re-tweet network as this network has well defined communities due to it capturing only the direct relationships between the people. LPA has the lowest modularity score on the Similarity based graph as the communities in this graph are overlapping and the connections too dense with many edges having the same weight.

Louvain Moduularity works the best on the Mention and the Re-tweet network as the algorithm itself tries to maximize the modularity value. In InfoMap, the best partition is expressed by minimizing the quantity of information needed to represent some random walk in the network. Since the Similarity based graph is dense, it requires minimum quantity of information to represent the random walk. Hence, the InfoMap method works better than Louvain Modularity in the Similarity based graph.

6 RELATED WORK

Twitter has become one of the main platforms of social activity. We have seen huge mobilizations over Twitter in recent uprisings. People have been engaging in using Twitter to express their views and opinions on various issues. Community detection is a fundamental task in social network analysis. A community can be defined as a group of users that interact with each other more frequently than those outside the group and are similar to each other than to the outside group. Twitter presents three types of connectivity information between users: follow, re-tweet and user mention. In the paper[6] it has been shown that re-tweets tend to happen between like-minded users rather than members of opposite camp.

In the paper [7] the authors consider that the word usage is the strongest indicator of user's orientation among all categories. The authors also incorporate user-word matrix and word similarity regularizer which provides the missing link in connectivity. In the paper [8] the authors present a statistical approach to analyze Twitter data using the concepts of social network generation and graph-based clustering. Tweets are tokenized using n-gram technique and analyzed using Latent Dirichlet Allocation (LDA) method to identify significant key terms, which are later on used for social network generation. Finally, Markov Clustering method is applied on the generated social network to identify dense regions (aka communities or cliques), each one representing the set of tweets related to a particular event. Tracking influence is another important task related to Twitter data analysis. Influential users play an important role in the society, and influence tracking may be useful for a number of applications ranging from election to marketing. In [5] the authors analyzed the role of structural features like in-degrees, re-tweets, and mentions for influence tracking and investigated the dynamic nature of user influence across topic and time. In [9] the authors apply clustering techniques (NFM) to analyze a Twitter network and obtain trends on Twitter

Graph	Measure	LPA	Louvain Modualrity	InfoMap
Mention network	Modularity	0.3945	**0.4708**	0.224
	Conductance	0.1462	0.0218	**0.0047**
Re-tweet network	Modularity	0.4486	**0.4803**	0.4147
	Conductance	0.1507	0.0611	**0.0071**
Similarity based network	Modularity	0.326	0.3284	**0.3901**
	Conductance	0.4011	0.2749	**0.0521**

Table 4: Evaluation of the Community Detection algorithms on the three graphs using Modularity and Conductance

7 CONCLUSION

In this project, we analyze the communities and trends of the #MeToo event on Twitter. We first create a Mention network, Re-tweet network and a Similarity based graph with tweets containing #MeToo posted between 25^{th} February'18 and 31^{st} March'18.

We have then implemented three community detection algorithms: LPA as the baseline, Louvain Modularity and InfoMap. After detecting the communities we have analyzed each community by looking into the user profiles and tweets to get a sense of the community.

The communities detected in the Mention and the Re-tweet network are different in their characteristics. The Re-tweet network was able to determine the direct connections between people. That is the communities in the Re-tweet network are exclusive in their membership and present only direct interactions. Whereas, the communities detected in the Mention network are much broader and contain implicit relationships. This is because in a Mention network, a person mentions other users in their tweets who they think have similar opinions/interests. By doing so, there are links formed that go beyond direct interactions.

We use conductance and modularity to evaluate detected communities which demonstrate that Louvain Modularity achieves highest modularity and relatively low conductance on mention graph and the re-tweet graph and InfoMap has the highest modularity and lowest conductance on similarity based graph. We have compared the three algorithms and have concluded that LPA works the best on a well defined network and InfoMap and Louvain work the best on a network where the communities are not well defined.

We also present a temporal analysis for the Mention network and the Similarity based graph. In Mention network, though the nodes in communities change over time, the characteristic of these communities remain unchanged. In Similarity based graph, the sentiment of largest community evolves corresponding to real-world events such as the Oscars and Women's day. It is interesting to see that the communities have specific traits over time. In order to understand the spread of this event we have also given a geographical representation of the tweets.

8 DIVISION OF WORK

The team collectively crawled the Twitter data using the Tweepy API and the code is available at our GitHub repository. By dividing the days between us, we were able to collect the data faster. Bhavika was responsible for the tweet text processing, the visualization of the Similarity based graphs and the communities in the Mention network, the geographical representation of the tweets and also analysis of the communities detected in Mention and Re-tweet network. She is also responsible for maintaining our Github repository. Cixing was responsible for creating the Similarity based graph using LDA, apply InfoMap community detection algorithm on the Similarity based graph and also do the community analysis. Siyan was responsible for creating the Mention network, Re-tweet network and writing the LPA algorithm, Louvain Modularity for the said graph. He visualized the LPA graphs and also analyzed the communities in re-tweet network. Vidya was responsible for raw tweet processing of extracting the required fields and also the processing the location field for the tweets. She computed the semantic results for the communities detected in the Similarity based graph, explored the community detection algorithms for Similarity based graphs and the methods to create Similarity based graphs. She worked on the temporal analysis of the sentiments on the similarity based graph and analyzed the communities detected in the Re-tweet network . All the four members mentioned their work in the final report.

ACKNOWLEDGEMENT

We would like to thank Professor Danai Koutra and Yujun Yan for their guidance and valuable inputs throughout the project.

REFERENCES

[1] Lambiotte Renaud Blondel Vincent D, Guillaume Jean-Loup and Lefebvre Etienne. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* (2008). http://10.1088/1742-5468/2008/10/P10008

[2] Michael I. Jordan David M. Blei, Andrew Y. Ng. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* (2003).

[3] Steve Harenberg, Gonzalo Bello, L. Gjeltema, Stephen Ranshous, Jitendra Harlalka, Ramona Seay, Kanchana Padmanabhan, and Nagiza Samatova. 2014. Community detection in large-scale networks: a survey and empirical evaluation. *WIREs Computational Statistics, Wiley* (2014). http://onlinelibrary.wiley.com/doi/10.1002/wics.1319/abstract

[4] J. Jiang J. Weng, E.P. Lim and Q. He. 2010. TwitterRank: Finding topic-sensitive influential twitterers. *International Conference on Web Search and Data Mining, pages 261fi270* (2010).

[5] F. Benevenuto M. Cha, H. Haddadi and K. P. Gummadi. 2010. Measuring user influence in twitter: The million follower fallacy. *Fourth International AAAI Conference on Weblogs and Social Media* (2010). http://hdl.handle.net/11858/00-001M-0000-0028-8BAB-9

[6] J. Ratkiewicz M. Francisco A. Flammini M. D. Conover, B. Goncalves and F. Menczer. 2011. Political polarization on Twitter. *5th InternationalConference on Weblogs and Social Media* (2011). https://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/viewFile/2847/3275

[7] Hasan Davalcu Mert Ozer, Nyunsu Kim. 2016. Community Detection in Political Twitter Networks using Nonnegative Matrix Factorization Methods. (2016). https://arxiv.org/pdf/1608.01771.pdf

[8] Muhammad Abulaish SMIEEE Nausheen Azamfk, Jahiruddin and Nur Al-Hasan Haldar. 2015. Twitter Data Mining for Events Classification and Analysis. *Second International Conference on Soft Computing and Machine Intelligence* (2015). http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7414678&tag=1

[9] Yong-Ho Ha Seongwon Lim Yong-Hyuk Kim, Sehoon Seo and Yourim Yoon. 2013. Two Applications of Clustering Techniques to Twitter: Community Detection and Issue Extraction. *Discrete Dynamics in Nature and Society, vol. 2013* (2013). http://hdl.handle.net/11858/00-001M-0000-0028-8BAB-9