Community Detection in Social Networks Facebook and Twitter

A Thesis Proposal
Presented to
the Faculty of the College of Computer Studies
De La Salle University Manila

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science in Computer Science

by

FERNANDEZ, Ryan Austin
POBLETE, Clarisse Felicia M.
SAN PEDRO, Marc Dominic
TAN, Johansson E.

Charibeth K. CHENG
Adviser

July 17, 2016

Contents

Chapter 1

Research Description

This chapter is an overview of the research undertaken in the field of community detection in social networks. This chapter is divided into four sections which are the current state of the technology, research objectives, scope and limitations, and significance of the research.

1.1 Overview of the Current State of Technology

Social media has become much more prevalent in recent years. People can now participate in what is called microblogging, a way for people to share their thoughts, status, and opinions in short posts, like Twitter, where posts are limited to one-hundred and forty characters (Java, Song, Finin, & Tseng, 2007). As such, these social media platforms are a prime opportunity to mine sentiments and to detect communities in the social network.

Community detection is clustering multiple users into groups where users within the group are more similar than users from outside the group (Tang & Liu, 2010). Community detection is necessary because it observes the interaction of multiple users as opposed to common mining methods which may only deal with local predictions i.e. predictions based on a single node as opposed to predictions considering the entire network of users.

Numerous studies on community detection have already been done. Zhang, Wu, and Yang (2012) defined features; such as text content similarity, URL similarity, hashtag similarity, following similarity, and retweeting similarity; that can be used to identify similarity between two nodes and aggregated these similarities

to detect communities. Their software just provided the listings of users in each community. K. Lim and Datta (2012) performed an inverted version in which they first defined interests and based on these interests, sought to extract communities from the network by identifying users that follow the top six celebrities, users with more than 10,000 followers, relating to the given interest. Their software only outputted the size and clustering coefficient of each community

Individual opinions of one specific user towards another was also studied by West, Paskov, Leskovec, and Potts (2014) combining sentiment analysis, using an L2-regularized logistic-regression classifier, with network analysis, inferring one users opinion of another by analyzing their common links with other users. Their output was the area under the curve of the receiver operating characteristic (AUC/ROC) and the precision-recall curves (AUC/negPR). In addition to these works, some visualizations have already been created such as SocialHelix by Cao, Lu, Lin, Wang, and Wen (2015), which uses the temporal extent of social communities, topics or events discussed, and user responses to topics and events to classify users into two sides of the argument. They then depict the two sides of an argument as strands in a double helix and their intersection defines events.

However, it is noticeable that all of these studies only involved Twitter and mainly used only common Twitter features for similarity analysis such as following networks, hashtag frequency, and retweet networks. There has yet to be a community detection tool that integrates data from Facebook, as well as the exclusive features from Facebook such as group membership and event membership, into the computation.

1.2 Research Objectives

1.2.1 General Objective

To produce a visualization of the detected communities on data found on Facebook and Twitter

1.2.2 Specific Objectives

1. To build a corpus of social media data;
2. To determine the various techniques and algorithms in detecting communities;

3. To determine the parameters/features to be used in detecting the communities;

4. To determine how to evaluate the correctness of the detected communities;

5. To implement a tool for the visualization of detected communities using the gathered information

1.3 Scope and Limitations of the Research

In order to perform a study on community detection, it is necessary to build a corpus of social media data. This research will cover searching for API's that will allow extraction of data from Facebook and Twitter.

Different techniques have been used in community detection. Among these techniques are the Infomap algorithm and the speaker-listener label propagation algorithm. This research will consider algorithms found in the review of related literature, including the Markov stability model, clique percolation method, k-means clustering, and divisive hierarchical clustering.

In the implementation of these community detection algorithms, it is necessary to identify which parameters/features/attributes indicate one users similarity to another. Inquiry will be done to identify these parameters, specifically how to extract them based on the raw data. The research will be limited to sentiment analysis and elements which can be extracted from a users post, which may include follow networks, hashtags, mentions, and retweets, which were mostly inspired from literature which focused on Twitter (Deitrick & Hu, 2013; Zhang et al., 2012; K. Lim & Datta, 2012). As such, Facebook specific features such as membership in groups and event participation may also be considered.

After community detection, it is necessary to determine whether the detected communities are sensible. Inquiry will be done to find appropriate metrics in determining the accuracy of detected communities. These algorithms will include average mutual following links per user per community or FPUPC (Zhang et al., 2012), modularity (Deitrick & Hu, 2013), and clustering coefficient (K. Lim & Datta, 2012).

After inquiring about multiple community detection algorithms and similarity parameters, it will be necessary to implement these algorithms in a working system. A visualization tool will be created that will only consider communities detected from data gathered from Facebook and Twitter.

1.4 Significance of the Research

Community detection is already a widely researched topic in the field of computer science. This research will contribute to that field by exploring a new domain, including Facebook in the scope of its community detection. This research can also contribute to the notion that community detection is a relevant field of study in this day and age.

This research can also be a very useful tool in the domains of viral marketing and political endorsement. This means that companies and governments may benefit from this research. Interested companies may use the result of this research to improve their sales and marketing. The government may use this to gauge public opinion on certain issues and to see analytics about which geographical areas have a particular opinion.

Chapter 2

Review of Related Literature

This chapter discusses the features, capabilities, and limitations of existing research, algorithms, or software that are related or similar to the proposed research. This chapter is divided into three sections. The first section discusses algorithms for community detection. The second section discusses algorithms for sentiment analysis and other similarity parameters used for community detection. The third section discusses community evaluation metrics.

2.1 Community Detection

Clauset, Newman, and Moore (2004) presents an improvement to community detection algorithms. Other existing algorithms presented in this are correct logically, but are computationally expensive and cannot run over extremely large datasets in a reasonable amount of time. This paper presents an algorithm that is identical in terms of output but is significantly faster than existing algorithms in terms of runtime.

The algorithm is based on the greedy optimization of modularity, which is a property that indicates the strength of division of a network into communities. The higher the modularity, the better the community structure. Given a graph with n vertices and m edges, the algorithm defines two quantities, namely:

$$e_{ij} = \frac{1}{2m} \sum_{vw} A_{vw} \delta(c_v, i) \delta(c_w, i) \tag{2.1}$$

which is the fraction of edges that join vertices in community i to vertices in community j, and

$$a_i = \frac{1}{2m} \sum_v k_v \delta(c_v, i) \tag{2.2}$$

which is the fraction of ends of edges that are attached to vertices in community i. The algorithm then defines the modularity Q as:

$$Q = \sum_i (e_{ii} - a_i^2) \tag{2.3}$$

The algorithm finds the largest Q that would result from merging two arbitrary communities, and merging those two communities. The algorithm's main offering is that it skips calculating Q when the two communities have no edges between them. offering an increase in performance. The algorithm maintains a matrix ΔQ_{ij} for each pair i, j of communities with at least one edge between them, a max-heap H containing the largest Q for each row of the matrix, and a vector array with elements a_i. ΔQ_{ij} is defined as follows:

$$\Delta Q_{ij} = \begin{cases} \frac{1}{2m} - \frac{k_i k_j}{(2m)^2} & \text{if i, j are connected,} \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

and a as:

$$a_i = \frac{k_i}{2m} \tag{2.5}$$

The actual algorithm runs as follows: (1) Calculate the initial values of ΔQ_{ij} and a_i, and populate the max-heap with the largest element of each row of the matrix ΔQ. (2) Select the largest ΔQ_{ij} from H, join the corresponding communities, update the matrix ΔQ, the heap H, and a_i, and increment Q by ΔQ_{ij}. (3) Repeat step 2 until only one community remains.

The proposed algorithm was run against purchase data from amazon.com. The data graph worked on was quite big, with 409 687 items and 2 464 630 edges. The algorithm was able to successfully structure the data into communities based mainly on purchasing info. The proponents were successful in discovering clear communities that correspond to specific topics or genres of books or music,

indicating that the co purchasing tendencies of Amazon customers are strongly correlated with subject matter.

The proponents hope that this algorithm will allow datasets with millions of vertices and tens of millions of edges to be processed using current computing resources in an efficient manner (Clauset et al., 2004).

Tang and Liu (2010) wrote a textbook on community detection. In the book, they sought to introduce characteristics of social media, review representative tasks of computing with social media, and illustrate associated challenges because with the emergence of social media websites, they felt it was an avenue to study human interaction and collaboration on an unparalleled scale. Multiple community detection approaches were discussed such as node-centric, network-centric, and hierarchy-centric.

Node-centric algorithms involve the maximum clique detection problem which involves searching for a maximum complete subgraph of nodes in a network graph that are all adjacent to each other. The clique percolation method can find overlapping communities by finding cliques of size k, and then producing a clique graph, wherein two cliques are connected if they share k-1 nodes. Each connected element in this clique graph is then a community.

Network-centric algorithms involves vertex similarity, which is the similarity of the nodes social circles or how many common friends the two nodes have. This is what structural equivalence deals with. Two nodes v_i and v_j are structurally equivalent if $\forall v_k \in \{x \mid x \neq v_i \wedge x \neq v_j\}\ e(v_i, v_k) \in E \iff e(v_j, v_k) \in E$, which is to say, v_i and v_j are connected to the exact same nodes. Nodes that are structurally equivalent belong to a community.

Hierarchy-centric algorithms come in two forms: divisive and agglomerative. In divisive clustering, the entire set of nodes starts out in one set and each time, each set is divided into two until each community only has one member. The division is done by finding the node with the lowest edge betweenness and removing it, since that node is most likely the node connecting two communities. Agglomerative clustering starts with each node in their own community and communities are joined if they increase the overall modularity of the set of communities. Modularity is given by

$$Q = \frac{1}{2m} \sum_{l=1}^{k} \sum_{i \in C_l, j \in C_l} (A_{ij} - \frac{d_i d_j}{2m}) \tag{2.6}$$

Where m is the number of edges, d_i is the degree of node v_i, C_l being the lth

community, and A_ij being the value in the adjacency matrix for node v_i and v_j.

These algorithms may be considered in the final community detection phase of the proposed research (Tang & Liu, 2010).

Lancichinetti, Radicchi, Ramasco, and Fortunato (2011) details an approach to performing community detection across a network. The authors of this paper argued that while there already exists a large variety of techniques for achieving this, there is still a need for more in-depth techniques that can handle different types of datasets, and the "subtleties" of community structure. This paper presents a technique called OSLOM (Order Statistics Local Optimization Method), which they claim to be the first method capable to detect clusters in networks accounting for edge directions, edge weights, overlapping communities, hierarchies and community dynamics. They claim that the algorithm performs just as well as other existing ones, and have been applied on several real networks. It is also freely available for anyone who wants to use it (Lancichinetti et al., 2011).

K. Lim and Datta (2012) aimed to detect communities that share common interests on Twitter, based on linkages among followers of celebrities representing an interest category because they wish to help markets identify target groups with common interests. However, their approach differs from the typical paradigm of "identify communities then, for each, identify interests", for they first identified interests they wish to extract communities from and from these interests, they then extracted the communities.

Given a set of celebrities, C, celebrities being users with more than ten thousand followers, the algorithm first gets the common followers of all the celebrities in the set using the formula.

$$P = \bigcap_{j \in C} (\bigcup_{i} link(i, c_j)) \tag{2.7}$$

Where $link(i, j)$ is given by

$$link(i, j) = \begin{cases} \{i\} & \text{i follows j} \\ \emptyset & \text{i does not follow j} \end{cases} \tag{2.8}$$

Given this set of fans, P, they used the Infomap Algorithm and the Clique Percolation Method to detect communities in P. For each interest they wish to extract communities for, they chose the top 6 most popular celebrities based on follower count. Google and Wikipedia were used to identify which interests a celebrity

represents. Afterwards, all users that follow the 6 celebrities were selected. They then selected 200,858 random users as a control group. Their algorithm produced more communities and larger communities than the control group, as well as more consistent communities, having a higher clustering coefficient.

In addition, the algorithm can be potentially applied to other social networking sites such as Facebook . In Facebook, celebrities could be represented by the Facebook pages of the different celebrities and the followership links can be defined as the different user "likes" on these pages (K. H. Lim & Datta, 2012).

This paper provides an interesting alternative to detect communities by first specifying the interest of the community before detecting it, which may be used in the proposed research on top of the usual algorithms (K. Lim & Datta, 2012).

Papadopoulos, Kompatsiaris, Vakali, and Spyridonos (2012) discussed methods of community detection in social media, comparing different aspects of specific methods, and discussing possible incremental applications of these methods. This survey aims to address two main elements left unaddressed in existing related survey articles, namely performance, in terms of aspects such as computational complexity and memory requirements, as well as the interpretation of results of community detection by social media applications.

Classes of community detection methods discussed include cohesive subgraph discovery, vertex clustering, community quality optimization, divisive, and model-based methods. Cohesive subgraph discovery comprises of methods that require the specification of certain structural properties that must be satisfied in order for a subgraph of the network to be considered a community. Vertex clustering makes use of traditional data clustering methods. Community quality optimization methods focus on the optimization of a graph-based measure of community quality. Divisive methods make use of identified edges and vertices in a network. Model-based methods consider dynamic processes or statistical models in the process of detecting communities.

The survey led to conclusions about the concept and structure of communities in the context of social media, to a rough classification of existing community detection methods, and to determining which methods are most appropriate for social media mining applications.

This survey provides an overview of existing methods of community detection as well as a comparison of these methods, which may aid the proponents in deciding the appropriate methods to use for the proposed research (Papadopoulos et al., 2012).

Xie (2012) details a study into two topics of social network analysis, namely opinion dynamics and community detection. In terms of community detection, one of the main difficulties presented is particularly challenging in large-scale networks. It presents an algorithm called Speaker-listener Label Propagation Algorithm (SLPA) for fast overlapping community detection. Another challenge presented was detecting communities in dynamic networks where changes happen often and in real-time, as with most real world applications. An algorithm for incrementally updating communities instead of profiling each snapshot is presented as well, called LabelRankT. This algorithm is claimed to drastically outperform existing detection algorithms, with similar results (Xie, 2012).

Zhang et al. (2012) sought to identify communities in Twitter based on common interests. This is because Twitter has become very popular recently but little is known of it in the user level. This study would help in user recommendation and tweet recommendation as well as viral marketing to specific target groups. To identify the communities, they first compute specific feature similarities, then aggregate these features to compute for the final user similarity, and then they used classical clustering algorithms to detect the communities. To identify the communities, they first compute specific feature similarities, then aggregate these features to compute for the final user similarity, and then they used classical clustering algorithms to detect the communities.

The specific features they used were textual contents. Each data point was the entirety of a users tweets. Latent Dirichlet Allocation was used to identify latent topics from the users tweets. URL similarity was also detected, finding which users share similar links. Hashtag similarity was also analyzed. The social structure of users was also analyzed, which includes following similarity and retweeting similarity.

In aggregating these similarities, the weighted sum of the previous similarities was computed to get the final similarity. Finally, k-means clustering was used to detect the communities based on their computed similarities.

This paper presents a possible framework for detecting the communities. The proposed research may even use similar features, in addition to the Facebook specific features, to detect similarity. The k-means clustering algorithm will be one of the proposed algorithms for use in the proposed research (Zhang et al., 2012).

Deitrick and Hu (2013) sought to use sentiment classification to analyze communities in Twitter because harvesting information from these online social networks (OSN) would aid in the fields of politics and marketing.

Their process is as follows: The follower network was represented as a weighted directed graph, each with initial weight of 1. To augment this, replies, mentions, retweets, hashtags, and sentiment classification of tweets were also harvested. These factors adjusted the weights in the graph. For community detection, the Infomap algorithm and Speaker-listener Label Propagation Algorithm(SLPA) were run.

Generally, the network with updated weights produced communities with greater modularity. Of the two algorithms, the Infomap algorithm performed better. Recurring sentiment analysis was also helped by performing the aforementioned algorithms on the accounts that have already been placed in detected communities, which permits more in-depth analysis into the users sentiment since it could be analyzed within the context of the detected community.

This research provides a new way to represent the network, with their updated weights, as well as more possible algorithms to consider in detecting communities (Deitrick & Hu, 2013).

Bakillah, Li, and Liang (2015) sought to contribute to the field of extracting relevant information from social media by detecting geo-located communities in Twitter in disaster situations. The main disaster they focused on is the occurrence of typhoon Haiyan in the Philippines.

Social graphs of Twitter users related to the focus are created by comparing Twitter's different interaction nodes like follow relations, mentions and tweet content. The fast-greedy optimization of modularity (FGM) clustering algorithm enhanced with semantic similarity is used in order to handle the complex social graphs created. Modularity measures the quality of divisions of a network into communities. By maximizing the modularity between the generated graph structure and a random graph structure, the optimal clustering results can be obtained. This modularity is expressed through the quality function Q:

$$Q = \sum_{c=1}^{n} \left[\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right] \tag{2.9}$$

where n is the number of clusters, m is the total number of edges, l_c is the total number of edges joining the vertices of cluster c and d_c is the sum of the expected random graph degree of the vertices of c. To achieve the largest quality change (ΔQ), communities are progressively merged. The usage of quality function Q and the merging of communities are the core steps of the FGM algorithm. Now, the FGM algorithm was enhanced by integrating graph based and text-based (text similarity measure) models to balance the importance of shared content

versus graph structure. This will be discussed more in detail in another section, Sentiment Analysis.

Together with FGM, the varied density-based spatial clustering of applications with noise spatial (VDBSCAN) clustering algorithm is used to get spatial communities at different time periods. This is done to divide thematic communities discussing same topics formed by the FGM algorithm into more meaningful sub-clusters. The discovery of geo-located communities could potentially help in identifying and locating incidents occurring during emergency situations.

The aim of the VDBSCAN spatial clustering algorithm is to find spatial clusters based on regions with higher density. The algorithm is based on two parameters:

1. r: the value of the radius that will be used to select members of a cluster
2. MinPts: minimal density to form a cluster.

Let D be the set of points corresponding to the geo-located tweets found in a given thematic community and that were issued during time period T.

The neighbour of point p is expressed as:

$$N(p) = \{q \in D | dist(p, q) \leq r\} \tag{2.10}$$

A cluster is generated based on the following properties, called density-reachable and density-connected:

1. A point q is directly density-reachable from a point p if

$$q \in N(p, r) and |N(p, r)| \geq MinPts \tag{2.11}$$

2. A point q is density-reachable from a point p if there exists a sequence of points p_1, , p_n where $p_1 = p$ and $p_n = q$ such that p_{i+1} is directly reachable from p_i, for all i.
3. A point q is density-connected to a point p if there is a point o such that p and q are density-reachable from o.

A cluster is a non-empty subset of D that satisfies the following properties:

1. $\forall$ p, q, if $p \in C$ and q is density - reachable from p, then $q \in C$ (maximality).

2. $\forall$ p, $q \in C$, p is density - connected to q (connectivity).

The parameters r and MinPts are optimized automatically based on the variation in density of the data set.

This research provides algorithms that could prove useful in getting the optimal clustering for detecting communities. It also gives an insight in considering the spatial and thematic properties of these communities (Bakillah et al., 2015).

Amor et al. (2015) sought to detect communities and to identify roles in the Twitter network regarding on the subject of the care.data debate using graph-theoretic methods, one of them being the Markov Stability method. There are two networks constructed from the obtained data relating to the care.data debate: follower network and retweet network. The flow-based community detection method Markov Stability was used for identifying interest communities in the follower network which resulted into a 13-way partition composed of 4 large communities and 9 minor ones. It is also used for the retweet network in order to find conversation communities which resulted into 8 communities.

The Markov Stability method works on the behaviour of dynamical processes on a network. This potentially reveals meaningful structure about the graph. It can extract different coarse-grained descriptions of the graph at different time scales. In addition, this method can find non-clique communities.

This research gives another community detection method that can be used for the proposed research. Its main advantage is its scalability especially over time scales (Amor et al., 2015).

Cao et al. (2015) proposed a visual analysis system, SocialHelix, because social media is a grand avenue for people to express their opinions and the researchers believed that an intuitive visualization that unfolds the process of sentiment divergence would have a far-reaching impact on multiple domains.

They first identified the key domain problems of social divergence before employing a data abstraction design to convert the raw data into a form that captures all the key factors of the aforementioned domain problems. This abstracted data is then represented in a visualization based on a visual DNA metaphor. In identifying the key domain problems, it is determined when divergences start and end, how they evolve, who is involved, what roles do they play, and why does divergence occur. In the abstraction phase, the raw data is decomposed into temporal extent of social communities, topics or events, and user responses to these topics or events. In the visualization phase, the opposite sides of the helix represent the two sides of a divergence. The helix curves represents the changes in the

communities sentiment. Nucleobase pairs represents events that connect the two communities.

In implementation, the data was first filtered, removing unrelated posts and people. Statistical Linguistic Sentiment Analysis was used to determine the users sentiment. Finally, clustering was done using Hadoop, producing a cluster with 30 nodes.

In the end, all test users were impressed by the visualization and agreed with the researchers model for the visualization. All test users felt that divergence identification was made easy due to the visualization.

This research gives a sample visualization that can inspire the proponents own visualization, albeit the goal is not to model single divergences but an entire community. A possible tool, Hadoop, is also mentioned, which may be used to cluster the data (Cao et al., 2015).

2.2 Similarity Parameters

This section outlines the basis/features/parameters used in detecting communities. It is divided into two subsections. The first subsection deals solely with sentiment analysis. The second subsection deals with other network and node parameters not related to sentiment analysis.

2.2.1 Sentiment Analysis

Zhang et al. (2012) provided a formula to determine similarity in terms of text.

$$sim_{text}(i,j) = \frac{1}{\sqrt{D_{js}(i,j)}} \tag{2.12}$$

D_{js} is the Jensen-Shannon Divergence between the two users topic probability distribution given by

$$D_{js}(i,j) = \frac{D_{kl}(UT_i \mid\mid M) + D_{kl}(UT_j \mid\mid M)}{2} \tag{2.13}$$

Where $M = \frac{UT_i + UT_j}{2}$ and $D_{kl}(P \mid\mid Q) = \sum_{i \in topics} P(i) \log \frac{P(i)}{Q(i)}$ and UT_i is the

probability distribution of user i for all topics. $UT_i[t]$ is the probability distribution for user i on topic t.

This research provides a metric to determine the similarity of two users in terms of post content, which can be used in the proposed research (Zhang et al., 2012).

Bryden, Funk, and Jansen (2013) aimed to determine whether or not members of identified communities had similar word usage and language features on their social media posts. This was done through the analysis of 75 million mutual tweets among 189 thousand Twitter users. This study focused on the connection of language has to network structure, in order to explore the potential of understanding society through analysis of communication on social media. Communities were characterized through the words used in messages sent by members of the community; the most representative words from each community were identified through the Z-scores of each words usage. The Euclidean distances between word usage frequencies for each pair of communities was the basis for determining how significant the differences between these communities word usages were. The research determined that there were many similarities in words, word fragments and word lengths among tweets from users in identified groups, including word usage that was not related to subject matter. Through language structure alone, the researchers were also able to determine a users' network communities.

This research focused on the detection of communities through language used on social media. As it involves on community detection on social media as well, the proposed research may make use of the approach presented in this research (Bryden et al., 2013).

Deitrick and Hu (2013) used a subjective/objective and positive/negative Naive Bayes classifier. To do this, all tweets were converted to lowercase; hashtags, usernames, urls were replaced with twitterhashtag, twitterusername, and twitterurl respectively; the tweet text was tokenized; repeated punctuation was replaced with the + sign e.g. !!! -¿ !+; sentence punctuation was split into separate tokens; non-sentence punctuation was removed. Ten-fold cross validation was used in training the classifier. Weights in the graph mentioned in section 2.1 were then updated if two users posted something with a similar sentiment and similar hashtag.

This research shows a clearly defined process in performing sentiment analysis, particularly the data cleaning step. This process could be adapted for the proposed research (Deitrick & Hu, 2013).

Bakillah et al. (2015) enhanced the FGM algorithm with a similarity measure.

A threshold T for text similarity is used to determine whether two communities are similar enough to increase the priority of merging them. 0.2-0.3 was used as the value of T. The cosine similarity measure is used to compute similarity between the communities' set of terms:

$$Cosinesimilarity = \frac{A \cdot B}{||A||||B||} = \frac{\sum_{k=1}^{l}(A_k \times B_k)}{\sqrt{\sum_{i=1}^{n}(A_i)^2 \times \sum_{j=1}^{m}(B_j)^2}} \tag{2.14}$$

A and B represent the frequency of a term in the set of terms associated with the first and second community, respectively. The similarity value ranges from 1, meaning dissimilarity, to 1, meaning exact similarity.

This measure can be used as a means for getting the similarity between different communities set of words when merging similar communities will be relevant to the proposed research (Bakillah et al., 2015).

2.2.2 Other Parameters

Zhang et al. (2012) provided a few formulas to determine similarity in terms of URL, hashtag, following, and retweeting similarity.

URL similarity is given by the same formula as text similarity in section 2.2.1, only using links instead of topics.

Hashtag similarity is given by

$$sim_{hashtag}(i,j) = \sum_{k=1}^{n}(1 - \left|\frac{N_{ik}}{|H_i|} - \frac{N_{jk}}{|H_j|}\right|)\frac{N_{ik} + N_{jk}}{|H_i| + |H_k|} \tag{2.15}$$

Where N_{ik} is the number of times user v_i used the hashtag k while H_i is the total hashtags used by v_i.

Following similarity is given by

$$sim_{follow}(i,j) = \frac{c_{friend}}{\sqrt{|Friend_i|\,|Friend_j|}} + \frac{c_{follower}}{\sqrt{|Follower_i|\,|Follower_j|}} \tag{2.16}$$

$|Friend_i|$ is the total number of users v_i follows. $|Follower_i|$ is the total number of users that follow v_i. c_{friend} represents the two users common friends. $c_{follower}$ represents the two users common followers.

Retweeting similarity is given by

$$sim_{retweet}(i,j) = \frac{c_{retweet}}{\sqrt{|R_i|\,|R_j|}} + \frac{n_{ij} + n_{ji}}{|R_i|\,|R_j|} \tag{2.17}$$

R_i is the number of users whom v_i retweet. $c_{retweet}$ is the number of users both v_i and v_j retweet. n_{ij} is the number of times v_i retweeted v_j and n_{ji} is the inverse case.

The aggregate similarity is now given by

$$sim(i,j) = \gamma_t sim_{text} + \gamma_u sim_{url} + \gamma_h sim_{hashtag} + \gamma_f sim_{follow} + \gamma_r sim_{retweet} \tag{2.18}$$

With $\gamma_{feature}$ determined in a process described in section 2.3.

This research gives formulas that can be used in the proposed research to measure similarity (Zhang et al., 2012).

Bakillah et al. (2015) created graphs with weighted edges with similarities based on Twitter's various interaction modes:

1. Graph based on mentions: A mention is a Twitter update that contains '@username' anywhere in the body of the tweet. It is used to reference another user. To build the graph, we create an edge between any pair of users u_1 and u_2 where u_1 issued a mention @u_2. The weight assigned to the edge increases with the number of mentions between u_1 and u_2 (nb_mentions(u_1, u_2) in the formula), but is normalized according to the total number of mentions in the graph and the total number of users:
$$Weight_{u_1u_2} = \frac{nb\_mentions(u_1,u_2)}{total\_nb\_mentions} \times total\_nb\_users \tag{2.19}$$
2. Graph based on follow relations: Edges are established between users linked by a follow relation. The weight assigned to the edge normalizes the importance of the follow relation according to the number of follow relations by user 1 and the average number of follow relations by user in the network:
$$Weight_{u_1u_2} = \frac{nb\_follow\_rel(u_1)}{total\_nb\_follow\_rel} \times total\_nb\_users \tag{2.20}$$

3. Graph based on shared URLs: Edges are created between any pair of users u_1 and u_2 who have shared the same URLs. The weight assigned to the edge increases with the number of URLs that u_1 and u_2 have shared (nb_URLs(u_1, u_2) in the formula), but is normalized according to the total number of shared URLs in the graph and the total number of users:

$$Weight_{u_1u_2} = \frac{nb\_URLs(u_1, u_2)}{total\_nb\_sharedURLs} \times total\_nb\_users \tag{2.21}$$

4. Graph based on similar Tweet content: Words common to different users tweets are used to build the graph. The resulting graph represents shared interests between users, but not necessarily an explicit relation between them. The tweet content are categorized according to corresponding text elements in the tweet. Edges are created between users whose tweets contain some common text elements. Edge weights are assigned according to the number of common text elements in tweets of users u_1 and u_2 and normalized according to the total number of text elements that were extracted from tweets:

$$Weight_{u_1u_2} = \frac{nb\_common\_text\_el(u_1, u_2)}{total\_nb\__text\_el\_extracted} \times total\_nb\_users \tag{2.22}$$

This study presents alternative formulae that may be used in the proposed research (Bakillah et al., 2015).

Amor et al. (2015) includes some sentiment analysis on the tweets they handled, particularly on negative tweets as these comprised most from sample they took. They divided the concerns of these negative tweets into 3:

1. Implementation - concerns regarding information provision, the opt-out process, and communication with the public
2. Scheme concept - concerns about privacy, sharing of personal data, and the use or sale of the data
3. Execution - Concerns around security, effectiveness of pseudonymisation, and cyber attacks

No formula or representation was given as to how tweets were categorized between these 3 concern categories. However, this opens up the idea of having specific parameters related to the focus of the community detection, in this case with regards to the care.data debate, instead of general parameters concerning the social sites interaction modes (Amor et al., 2015).

Darmon, Omodei, and Garland (2015) aimed to present an approach to community detection that is multifaceted, focusing not only on structure-based communities, but on other types as well, namely activity-based, topic-based, and interaction based communities. Communities can be defined similarly or differently according to these types, so in order to come up with a more accurate and dynamic picture of a community, all types of communities, as well as the overlaps among these communities, should be taken into account. This study was done through the analysis of a Twitter dataset in order to assign representative weights for each community type. Activity-based communities were derived through the timing of users' tweets, topic-based communities were derived from hashtag similarities, and interaction-based communities were derived from retweets and mentions.

For activity-based communities, at time t, a user u either posts a tweet ($X_t(u) = 1$) or does not post a tweet ($X_t(u) = 0$); this is how each users behavior is viewed. The flow of information from a user u to a follower f is represented by the estimated transfer entropy between their time series $X_t(u)$ and $X_t(f)$, which is computed through the following formula.

$$W_{u\to f}^{TE(k)} = \tilde{TE}_{X(u)\to X(f)}^{(k)} \tag{2.23}$$

For topic-based communities, edges on the network of users and followers are weighted depending on the number of common hashtags between each user and follower pair. This weight is calculated using the following formula.

$$W_{u\to f}^{HT} = \frac{\vec{h}(u)\vec{h}(f)}{\left|\left|\vec{h}(u)\right|\right| \left|\left|\vec{h}(f)\right|\right|} \tag{2.24}$$

Interaction-based communities are defined by three weighting schemes. The first scheme considers the number of tweets follower f retweeted from user u.

$$W_{u\to f}^{R} = \frac{\#\text{ retweets of } u \text{ by } f}{\#\text{ total retweets made by } f} \tag{2.25}$$

The second scheme considers the number of tweets wherein user u mentions follower f.

$$W_{u\to f}^{R} = \frac{\#\text{ mentions of } u \text{ by } f}{\#\text{ total mentions of } f} \tag{2.26}$$

The third and final scheme takes the arithmetic mean of the mentions and retweets.

The study determined that the multifaceted approach to community detection could aid in better understanding the structure of online communities and in finding communities in social media that would otherwise be hidden.

The study provides an approach that may be considered as well as algorithms that may be used in the detection of communities in the proposed research (Darmon et al., 2015).

2.3 Community Evaluation Metrics

Zhang et al. (2012) used the average number of mutual following links per user per community(FPUPC) to evaluate their communities. Based on this, appropriate weights for the aggregation were found by first performing their k-means clustering algorithm using only one feature similarity for each of the similarities and extracting the FPUPC. Afterwards, they gave each feature similarity a weight based on the following formula.

$$\gamma_{feature} = \frac{FPUPC_{feature}}{\sum_{f \in features} FPUPC_f} \tag{2.27}$$

The number of clusters, k, used in the k-means clustering algorithm was also tweaked to get the maximum FPUPC. They concluded that they were successful in generating relatively accurate communities due to the incrementally increasing FPUPC after adjusting the weights.

This provides one possible evaluation metric for the proposed research, as well as a method to provide weights for the feature similarities that the proponents will eventually be using for community detection (Zhang et al., 2012).

Table 2.1: Summary of Review of Related Literature

Reference	Community Detection Algorithms	Sentiment Analysis Model	Other parameters	Community Evaluation
(Clauset et al., 2004)	Greedy Optimization of Modularity			
(Tang & Liu, 2010)	Clique percolation method, similarity detection, divisive and agglomerative clustering			
(Lancichinetti et al., 2011)	Order Statistics Local Optimization Method			
(K. H. Lim & Datta, 2012)	Topic driven community detection, Infomap method, Clique percolation method			

(K. Lim & Datta, 2012)	Topic driven community detection, Infomap method, Clique percolation method			
(Papadopoulos et al., 2012)	Comparison of Existing Methods			
(Xie, 2012)	Speaker-listener Label Propagation Algorithm, LabelRankT		Correlations between different snapshots of the network over time	
(Zhang et al., 2012)	k-means clustering	Similarity Formula for Text	Similarity Formula for URL, Hashtag, Follower, and Retweeting	FPUPC metric
(Bryden et al., 2013)		Characterization of communities through word usage		

(Deitrick & Hu, 2013)	Weighted directed graph, Infomap Algorithm, SLPA	Subjective/Objective, Positive/Negative Naive Bayes Classifier	replies, mentions, retweets, hashtags	
(Bakillah et al., 2015)	enhanced fast-greedy optimization of modularity (FGM) algorithm with similarity measure, varied density-based spatial clustering of applications with noise spatial (VDBSCAN) algorithm	cosine similarity measure	mentions, follow relations, shared URLs, Tweet content	
(Amor et al., 2015)	Markov Stability		care.data debate - implementation, scheme concept and execution	

(Cao et al., 2015)	Data abstraction design, Hadoop tool	Temporal extent of posts, topics and events, user responses, Statistical Linguistic Sentiment Analysis		
(Darmon et al., 2015)			Activity-based communities, Topic-based communities, Interaction-based communities	

References

Amor, B., Vuik, S., Callahan, R., Darzi, A., Yaliraki, S. N., & Barahona, M. (2015). Community detection and role identification in directed networks: understanding the twitter network of the care.data debate.

Bakillah, M., Li, R.-Y., & Liang, S. H. L. (2015, February). Geo-located community detection in twitter with enhanced fast-greedy optimization of modularity: The case study of typhoon haiyan. *Int. J. Geogr. Inf. Sci.*, *29*(2), 258–279. doi: 10.1080/13658816.2014.964247

Bryden, J., Funk, S., & Jansen, V. A. (2013). Word usage mirrors community structure in the online social network twitter. *EPJ Data Science*, *2*(1), 1–9. doi: 10.1140/epjds15

Cao, N., Lu, L., Lin, Y.-R., Wang, F., & Wen, Z. (2015). Socialhelix: visual analysis of sentiment divergence in social media. *Journal of Visualization*, *18*(2), 221–235. doi: 10.1007/s12650-014-0246-x

Clauset, A., Newman, M. E. J., & Moore, C. (2004, Dec). Finding community structure in very large networks. *Phys. Rev. E*, *70*, 066111. doi: 10.1103/PhysRevE.70.066111

Darmon, D., Omodei, E., & Garland, J. (2015, 08). Followers are not enough: A multifaceted approach to community detection in online social networks. *PLoS ONE*, *10*(8), 1-20. doi: 10.1371/journal.pone.0134860

Deitrick, W., & Hu, W. (2013). Mutually enhancing community detection and sentiment analysis on twitter networks.

Java, A., Song, X., Finin, T., & Tseng, B. (2007, August). Why We Twitter: Understanding Microblogging Usage and Communities. In *Procedings of the joint 9th webkdd and 1st sna-kdd workshop 2007* (p. 56-65). Springer.

Lancichinetti, A., Radicchi, F., Ramasco, J. J., & Fortunato, S. (2011). Finding statistically significant communities in networks. *PLoS ONE*, *6*(4), 1 - 18.

Lim, K., & Datta, A. (2012). Following the follower: Detecting communities with common interests on twitter. In *Proceedings of the 23rd ACM conference on hypertext and social media (ht12)* (Vol. 1, pp. 317–318). Association for Computing Machinery. doi: 10.1145/2309996.2310052

Lim, K. H., & Datta, A. (2012). Finding twitter communities with common

interests using following links of celebrities. In *Ht*.

Papadopoulos, S., Kompatsiaris, Y., Vakali, A., & Spyridonos, P. (2012). Community detection in social media. *Data Mining and Knowledge Discovery*, *24*(3), 515–554. doi: 10.1007/s10618-011-0224-z

Tang, L., & Liu, H. (2010). *Community detection and mining in social media*. Morgan & Claypool. doi: 10.2200/S00298ED1V01Y201009DMK003

West, R., Paskov, H. S., Leskovec, J., & Potts, C. (2014). Exploiting social network structure for person-to-person sentiment analysis.

Xie, J. (2012). *Agent-based dynamics models for opinion spreading and community detection in large-scale social networks* (Unpublished doctoral dissertation). Troy, NY, USA. (AAI3533361)

Zhang, Y., Wu, Y., & Yang, Q. (2012). Community discovery in twitter based on user interests. *Journal of Computational Information Systems*, *8*(3), 991–1000.