Understanding the User Preferences Using Social Network Mining

A Dissertation submitted in the partial fulfillment for the award of

MASTER OF TECHNOLOGY

IN

SOFTWARE TECHNOLOGY

by

Atul Mittal

Roll no. 2K11/SWT/04

Under the Essential Guidance of

Mr. Manoj Kumar

Associate Professor

Department of Computer Engineering

Delhi Technological University

New Delhi

2011-2014

DECLARATION

I hereby declare that the thesis entitled "**Understanding the User Preferences Using Social Network Mining"** which is being submitted to the **Delhi Technological University**, in partial fulfillment of the requirements for the award of degree of **Master of Technology in Software Technology** is an authentic work carried out by me. The material contained in this thesis has not been submitted to any university or institution for the award of any degree.

_____

Atul Mittal
Department of Computer Engineering
Delhi Technological University,
Delhi.

CERTIFICATE

DELHI TECHNOLOGICAL UNIVERSITY

(Govt. of National Capital Territory of Delhi)

BAWANA ROAD, DELHI-110042

Date: _____

This is to certify that the thesis entitled **"Understanding the User Preferences Using Social Network Mining"** submitted by **Atul Mittal (Roll Number: 2K11/SWT/04),** in partial fulfillment of the requirements for the award of degree of Master of Technology in Software Technology, is an authentic work carried out by her under my guidance. The content embodied in this thesis has not been submitted by her earlier to any institution or organization for any degree or diploma to the best of my knowledge and belief.

Project Guide

Mr. Manoj Kumar

Associate Professor

Department of Computer Engineering

Delhi Technological University, Delhi-110042

ACKNOWLEDGEMENT

I take this opportunity to express my deepest gratitude and appreciation to all those who have helped me directly or indirectly towards the successful completion of this thesis.

Foremost, I would like to express my sincere gratitude to my guide **Mr. Manoj Kumar**, **Associate Professor, Department of Computer Engineering**, **Delhi Technological University, Delhi** whose benevolent guidance, constant support, encouragement and valuable suggestions throughout the course of my work helped me successfully complete this thesis. Without her continuous support and interest, this thesis would not have been the same as presented here.

Besides my guide, I would like to thank the entire teaching and non-teaching staff in the Department of Computer Science, DTU for all their help during my course of work.

ATUL MITTAL

Roll No. 2K11/SWT/04

Table of Contents

List of figures

ABSTRACT

Recommender systems have become an important part of the web sites; the vast number of them is applied to e-commerce. They help people to make decision, what items to buy, which news to read, or which movies to watch. Recommender systems are particularly useful in environments with information overload since they cope with selection of a small subset of items that appear to fit the user's preferences.

The global network provides a vast amount of diverse data useful for social network analysis, e.g., for the estimation of the user social position or finding significant individuals or objects. Internet-based social networks can be either directly maintained by dedicated web systems like Twitter, Facebook , LinkedIn or extracted from data about user activities in the communication networks like e-mails, chats, blogs, homepages connected by hyperlinks , etc. Some researchers identify the communities within the Web using link topology, while others analyze the e-mails to discover the social network.

Based on semantic web analysis and using soft computing techniques and data mining tools the relevant information is obtained from the social network and by applying the different techniques and approaches of data mining and soft computing, data can be clustered to be an input to the Recommender system. The main focus of this thesis is extracting the relevant information from the social network site like Twitter using SNM and designing an appropriate RS for understanding the user preferences.

CHAPTER 1
Introduction

The current phase on the internet is witnessing a tremendous growth of social networks and huge amounts of new data are being created every second. With the advent of social networks, it has also become possible to disseminate this information at very fast rates. Millions of new user posts everyday are being created on social networking sites like Facebook1, Twitter2, Wordpress3 and Flickr4. In this section, I present a brief introduction about social networks with a special focus on twitter. In this report, I will be describing about our experiments on real data collected from twitter from September, 2011 to January, 2012. Twitter is not only a fantastic real-time social networking tool; it also acts as a great source of rich information for data mining. On an average, the users on twitter produce more than 140 million5 tweets per day (March 2011). This section introduces concepts of social media followed by specific twitter lingo and finally presents a brief overview of the past researches in this field.

1.1. Social Media

Social Media has recently evolved into a source of social, political and real time information. In addition to this it is also a great means of communication and marketing. People have been sharing information on social networks through the use of status updates , blogging, sharing multimedia content like images and videos as well as interacting together thereby forming groups and communities on social networks. Monitoring and analyzing this information can lead to valuable insights that might otherwise be hard to get using conventional methods and media sources. The social networking sites such as Facebook, Twitter and Flickr provide a new way to share the information among them and get frequent updates. In addition to this, the sites also allow sharing of additional information which can be important in analyzing the contents, e.g. location etc.

The social media has an advantage over conventional media sources as it is managed by the users. Conventional media only allowed users to gain information that was provided to them. The flow of information was only one-sided from the media to user. With social networks, however, the users now have the ability to respond to the news and events around them and provide their opinion on them as well as share them. This leads to the evolution of a multi way mode of information dissemination in which the users post information along with other information like links, images and videos. As a result, a user generated model of information is generated. The social graph of users and their connections on the social networks plays an important role in analyzing this information model in order to obtain meaningful data from the vast amount of "user generated content" that is created every day. Since, the micro-blogging6 sites like Facebook, Twitter and Flickr allow users to share short messages and multimedia, they have become an instant source of information through which users from all around the world can remain connected and get to know about the information from several sources.

1.2. Twitter

Twitter launched as a micro-blogging website in March 2006 which allows users to post status updates of up to 140 characters, also known popularly as tweets. Since its launch, twitter has amassed a large user base and now has over 300 million users (June, 2011) .Twitter allows its users to post short status messages called tweets. Tweets can be posted (tweeted) from various sources which include the twitter website, twitter mobile applications as well as several third party applications/websites (after authentication). Users also have the control over the privacy features and they can choose to either make their tweets public which make the tweets visible to any one or make them private which restricts the access to only some users who obtain permission from the user. Users can follow other users on twitter which gives them access to their tweets on their homepage on twitter.

Twitter allows several other features. It allows users to reply to tweets of other users by clicking on the reply button on the tweet of the user who one wants to reply to. This is a way to say something back in response to a user's tweet. In addition to this, users can also mention other

users in their tweets by adding '@' to the username of another user in a tweet. A mention is a way to refer to some other user. Another popular concept of twitter is re tweeting. A re tweet is an event of sharing someone else's tweet to our followers. Re tweet plays an important part in the dissemination of information on twitter. Users can also add a hash tag in their tweets by adding a '#' sign before relevant keywords. This is used to categorize those tweets to show more easily in twitter search. Very popular hash tags on twitter become trending topics on twitter.

An important feature of twitter that separates it from other social networking sites like Facebook is that the relationship of following and being followed are not necessarily two ways. Following someone is equivalent to subscribing to a blog; the follower gets all the status updates of the user that he follows.

An important characteristic that emerges from the network of twitter users is the Social Graph. A social graph is a graph derived from the connections between the users. These connections can be of many forms. The most straightforward social graph that can be created from twitter is a graph that contains following and being followed relationship among users. There have been several researches focused towards studying these social graphs and finding some features from such graphs. There are a few properties common to many social graphs: the small-world property, power law degree distributions and network transitivity (two users who have a common neighbor are more likely to be connected together rather than with some other user who with whom they don't share a neighbor). The social graphs generally also contain a clustered structure meaning that certain users' form a tightly knit group with very low connectivity between different such groups. These clusters may also contain other similarity features like similar tweets or locations etc. A community in a social graph can be described as a group of vertices that have more edges between them than any other vertex that belongs to other group in the social graph.

1.3. Communities in Social Networks

The topology of complex social networks has been studied extensively in the past. It has been found that social networks exhibit a very clear community structure. This community structure can occur due to personyl as well as political or cultural reasons. The analysis of community structure on social networks can be used to figure out influential tweets and user groups for specific brands, sports, political organizations and technologies.The communities have also been analyzed to discover disaster events etc.

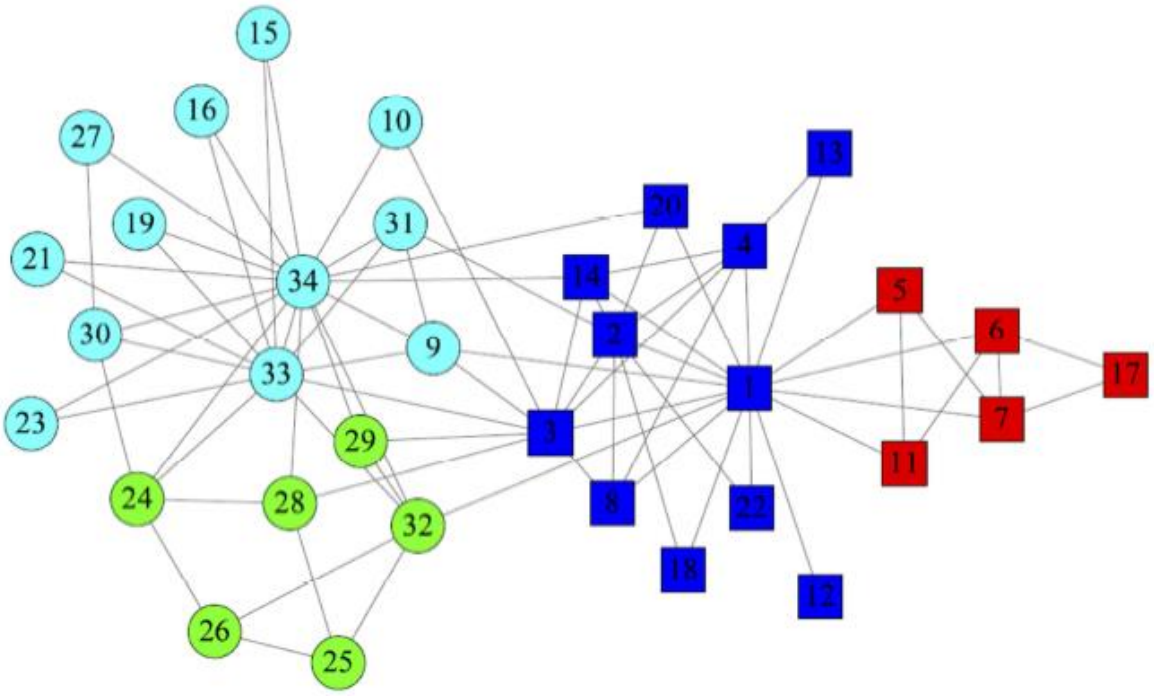

Figure 1: The Karate Club network

Figure 1 presents an example of a traditional network, Zachary's karate club network which has been widely used to evaluate community structure and detection in networks. The network shows social interactions between individuals at a karate club at an American University. The club split into two groups as a result of a dispute between club's administrator and principal karate teacher. The real social structure in the graph is shown by squares and circles depicting the group of individuals who sided with the administrator and others with the karate teacher. However, there

have been several researches and community detection methods have also come up with another meaningful clustering result as shown by different colors in the graph.

Majority of the algorithms for social network analysis only consider the social connections between users for the analysis of clusters between users and ignore the vast amount of other information available in the current social networks. In addition to social connections, twitter can be used to obtain different types of links among users like mentions, similarity between tweets of different users, re tweets, hash tags and locations.

I analyze several different clustering algorithms by using different link structure between users by taking into account the social connections, mentions, hash tag similarity, tweet similarity etc. among users. I also analyze two different types of algorithms, firstly where I know the ground truth data and therefore the number of clusters, and then, the algorithms that allow clustering without using the information about number of clusters.

1.4. Goal of the thesis

The goal of the work in this thesis is summarized below:

a. ***To collect the tweets data from the Twitter website-*** As discussed earlier, the main problem is to get the data from the social network site like Twitter, Facebook etc. In this thesis I will extract the user tweets data from the Twitter website for future analysis purpose.

b. ***To search #keywords from the tweeter's tweets*** **–** Our aim to search the relevant keywords which are present with hash tag (#) in tweets data. On the basis of those keywords, we can extract the tweets from tweeter web site.
c. ***To search based on Geo code-*** As twitter can be search on the basis of geographical code. Tweets twitted on the particular location can be identified and these will be helpful to take the popularity decision on the basis of different –different locations.

d. ***To write the sentiment function-*** Our main aim of this thesis is to compute the popularity of the searched keyword. This can be computed by writing the sentiment function which will check the positive and negative words used in the tweet. And based on the count of positive and negative words we will get the popularity results.

e. ***Scoring tweets based on the sentiment function-*** Sentiment function which will check the positive and negative words present in the tweet. And based on the count of positive and negative words we will get the score of that tweet.

f. ***Visualization by Histogram-*** Tweet score can be represented by the histogram. Histogram is the good way to represent the computed figure.

g. ***Comparison of data sets-*** We can compare the different data sets by comparing the histogram of the different data sets.

h. ***Graph representation-*** By using the above mentioned tweet data, we can represent that data in graphs, and Graphs are very useful in the social network analysis.

1.5. Organization of Thesis

This thesis is organized as follows:

Chapter 1 presents a detailed introduction about the background of social media, twitter and related topics.

Chapter 2 discusses the previous work done in the field of social network mining. This includes the extensive study of various approaches and techniques that have been proposed in the literature so far. It also highlights some of the most relevant works in the direction of field of work presented in the thesis

Chapter 3 I begin by presenting an overview of our data collection system that includes the environment that I use for collecting data. In addition to this, I also describe more about the filters using which I collect our data in this chapter.

Chapter 4 presents a detailed analysis of the results obtained and a top level analysis of different types of collected data and present visualizations.

References

Appendix 1 source code

CHAPTER 2

Literature Survey

Salton has defined that information retrieval as a field that takes into account the structure, analysis, management, storage, search and access to the information. The main purpose is to facilitate users in retrieving information through the information retrieval system. Users must translate their information needs in the form of queries that can be processed by the information retrieval system.

Information filtering technique plays an important role in developing a recommender system. Belkin and Croft view information filtering as a type of information retrieval systems. The aim of information retrieval and information filtering is to select relevant information and convey them to the users. The objective of information filtering is to remove irrelevant data from the flow of data items. Profiles are used as additional knowledge for queries to search, collect and transmit relevant information to users.

Collaborative filtering is a technology used to automate the process of the human recommendation. It is the most successful technology for the recommendation system and has developed and improved for decades. This technique concerns with finding groups that have similarities from a large number of groups. At first, the recommender system has been known by the term 'collaborative filtering' because this technique is used as an algorithm to develop the recommendation system. Resnick and Varian later in their study entitled 'recommender system' has suggested a more general definition. They defined the recommendation system as a system that takes into account the individual recommendations as input, classify, aggregate and send them to the appropriate users.

In some prior work, researchers mainly focused on the usage pattern and network properties of Twitter, to figure out the problem “what is Twitter and how do people use Twitter”. For example, Java et al. studied the topological and geographical properties of Twitter’s social network and presented a brief taxonomy of user intentions as information sharing, information

seeking and social activity. Honeycutt and Herring analyzed the usage of @ symbol to measure the conversation and collaboration on Twitter.

Their findings discovered a high degree of conversational engagement. Zhao and Rosson qualitatively investigated the motivation of using Twitter and explored its potential impacts on informal communication. All of them have revealed that Twitter was mainly used in two different ways: as an information platform or as a social network. Recently, with the rising popularity of Twitter, more research in a number of areas has been spurred to leverage its great wealth of both textual and social information. For instance, Twitter has been discussed to discover breaking news, detect natural disasters, improve real time web search, characterize media events and identify influential users or interesting content.

While most of the published research we mentioned above has focused on the network structure or the textual content of Twitter, less work has presented a systematic modeling of users' topics of interest on Twitter. Weng et al. collected the tweets published by individual user into a big document and used LDA to discovery his latent topics of interest. Chen et al. compared two different bag-of-words profiles for each Twitter user and found that profile built on user's own tweets worked better than on his followers' tweets. Michelson and Macskassy categorized the entities in the tweets by leveraging Wikipedia as a knowledge base and built a topic profile for each user on those categories. Most of these works have aggregated posts of the same user to build his topic profile, without considering the social networking function of Twitter. To filter out noise, some of them adopt word-level selections, such as to remove words with low tf-idf scores or low frequent categories. We believe that in comparison with those word-level elections, a tweet level selection to distinguish tweets between user interest and social activities will capture the real motivation of tweets, thus can reach a better understanding of users' topics of interest.

Among the several published works on Twitter summarization, Sharifi *et al.* find important phrases to be included in a summary with a graph-based algorithm, but the authors later develop a simpler "Hybrid TF-IDF" method, which ranks tweet sentences using the TF-IDF scheme and produces even better results. This is also confirmed by Inouye, who shows that Hybrid TF-IDF

outperforms several other main stream summarization approaches, including MEAD , Lexmark ,and Text Rank . Amore complicated work is reported by Liu *et al*, which highlights the use of linked webpage content and relies on Integer Linear Programming-based optimization to extract tweet sentences.

In the past few years, much research has focused on analyzing Twitter users and the content they produce. Abel et al. Investigated hash tag-, entity- and topic-based Twitter user profiles for personylized news recommendations. They found that user activity on Twitter was correlated with semantics extracted from news articles. Hannon et al. used tweet content (from the ego and the alters) and collaborative filtering (neighbor's IDs) approaches to create user profiles and recommend Twitter users to follow. They ranked relevant users by a search engine based on a target user profile or relevant query terms. In the context of Twitter user clustering, there are also some works that study the use of both content and social structure. Zhang et al. Investigated the problem of Twitter user clustering based on user interests. They calculated user similarity by linearly aggregating five different user similarity approaches which included similarity of tweet text, URLs, hash tags, follower relationship and re tweeting relationship.

Karandikar demonstrated a topic model based k-means algorithm for user clustering on Twitter. The author first generated a topic vector which yielded a distribution over each topic for every user. He then used k-means to cluster users based on the topic vector of each user.

CHAPTER 3

System Design and Data Collection

In this section I present a brief overview of the design of our system for data collection and the experimental setup and filters based on which I collect our data. I follow this brief description by the goals of the data collection system.

3.1. System Architecture

Figure 2 describes the scope of the system that I built along with the entities outside the system that it interacts with and a description of the interfaces between these entities. Although the early web was about human-machine interaction, today's web is about machine-machine interaction, enabled using web services. These services exist for most popular websites—from various Google services to LinkedIn, Facebook, and Twitter. Web services create APIs through which external applications can query or manipulate content on websites. Twitter API7 provides interfaces to query Twitter for data based on certain filters.

Each API represents a facet of Twitter, and allows developers to build upon and extend their applications in new and creative ways. Twitter provides three kinds of APIs:

- **Search API**

The Search API allows users to query for Twitter content. This includes finding tweets for a set of keywords, users or location posted in the past.

- **REST API**

The REST API enables developers to access some of the core primitives of Twitter including timelines, status updates, and user information. In addition to offering programmatic access to the timeline, status, and user objects, this API also enables developers a multitude of integration opportunities to interact with Twitter.

- **Streaming API**

The Streaming API allows for large quantities of keywords to be specified and tracked, retrieving geo-tagged tweets from a certain region, or have the public statuses of a user set returned. It allows users to establish and maintain a long lived HTTP connection with the Twitter server.

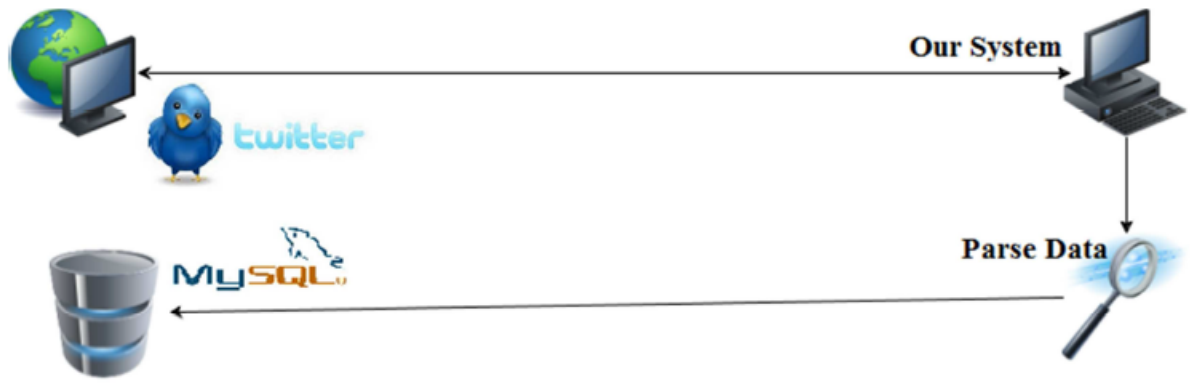

Figure 2: System Architecture

I now present in Figure 3 the schema of our database that represents the information that I collect for the users and tweets. The user table contains the information for the users on twitter. I contain all the available profile information from twitter for the users. This enables us to utilize different kinds of information other than just the social connections for the clustering algorithms. The status table contains various information related to tweet posted by user. The text attribute contain the actual text posted by a given user. In addition to this it also contains the longitude and latitude of the location from where tweet has been posted, time etc. It also contains a link to the user table which points to the user who has posted a particular tweet. I also store the information about the place from which the tweet was posted in the place table. This allows us to collect all tweets that have been collected from the same place easily. The tweets can also contain several hash tags, mentions, links, and images which are stored in the respective tables in the database. I store these information separately in different tables so that I don't have to perform text manipulation to obtain these information from the database.

Figure 3: Database Schema for Data collection

3.2. Technologies Used

In this section I present a brief overview about the technologies used for building the system for collecting data as well as preparing further analysis.

- **Eclipse:** I use Eclipse for the development. The reason for selecting Eclipse is its wide community. Most of the plug-in can be easily integrated with it.

- **Maven:** Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model, Maven can manage a project's build, reporting and documentation from a central piece of information. Maven uses a construct known as a Project Object Model to describe the software project built, its dependencies on other external modules and components, and the build order. It comes with pre-defined targets for performing certain well defined tasks such as compilation of code and its packaging.

- **Hibernate:** Hibernates primary feature is mapping from Java classes to database tables (and from Java data types to SQL data types). Hibernate also provides data query and retrieval facilities. Hibernate generates the SQL calls and attempts to relieve the developer from manual result set handling and object conversion and keep the application portable to all supported SQL databases with little performance overhead. Hibernate is basically an ORM Framework which allows you to perform database activities without bothering about the Database change. With respect to performance, hibernate provide the capability to reduce the number of database trips by creating the batch processing and session cache and second level cache. It also supports the transactions. More than this all, it is very easy to make a cleaner separation of Data Access Layer from Business logic layer. With all the capabilities mention above it is fast and easy to learn hibernate, develop application and maintain easily. The core drawback of JDBC is that it doesn't allow you to store object directly to the database you must convert the objects to a relational format.

- **Twitter:** It is an open-source, Google App Engine safe Java library for Twitter API which is released under BSD license. It allows to easily integrating a Java application with the twitter service. I have used it to collect tweets using its streaming and search methods implementation from the twitter4j package.

- **Git and Github:** Git is a distributed revision control system. Every Git working directory is a full-fledged repository with complete history and full revision tracking capabilities, not dependent on network access or a central server. GitHub is a web based hosting service for software development projects that use the Git revision control system. Git Hub offers both commercial plans and free accounts for open source projects. I use git for version control and Github to manage code between different systems and developers.

- **Matlab:** Matlab can be used for the implementation of various clustering algorithms as it provides several tools for the analysis of matrices in which the social connection graphs can be represented easily.

- **R Studio:** RStudio is a free and open source integrated development environment (IDE) for R, a programming language for statistical computing and graphics. RStudio is available in two editions: RStudio Desktop, where the program is run locally as a regular desktop application; and RStudio Server, which allows accessing RStudio using a web browser while it is running on a remote Linux server. Prepackaged distributions of RStudio Desktop are available for Microsoft Windows, Mac OS X, and Linux.

- **Gephi**: Gephi is an open-source network analysis and visualization software package written in Java on the NetBeans platform, initially developed by students of the The University of Technology of Compiègne (*Université de Technologie de Compiègne* or UTC) in France. Gephi has been used in a number of research projects in the university, journalism and elsewhere, for instance in visualizing the global connectivity of *New York*

Times content and examining Twitter network traffic during social unrest along with more traditional network analysis topics.

3.3. Data Collection

I collect different type of data in order to fulfill different goals. In this section, I provide a brief description of the data that I collect using our system followed by the objectives that the collected data helps to achieve. I collect data of the following three types:

3.3.1 Geo-tagged tweets

Twitter's Tweet with Your Location feature allows users to selectively add location information to their Tweets. The users who choose to add location to their tweets will be able to add their location information to new tweets that they post. Some applications allow users to tweet with their exact geo-location coordinates of the location from which they tweet. Figure 4 shows an example of a geo-tagged tweet posted on twitter. I collect tweets that come from the following five cities:

- London: (51.3695, -0.3475)15 to (51.6435, 0.0915)
- New York: (40.633¸-74.11) to (40.800, -73.89)
- Paris: (48.784, 2.241) to (48.929, 2.2460666)
- San Francisco: (37.6925, -122.529) to (37.8661, -122.3094)
- Mumbai: (18.875, 72.55) to (19.275, 73.15)

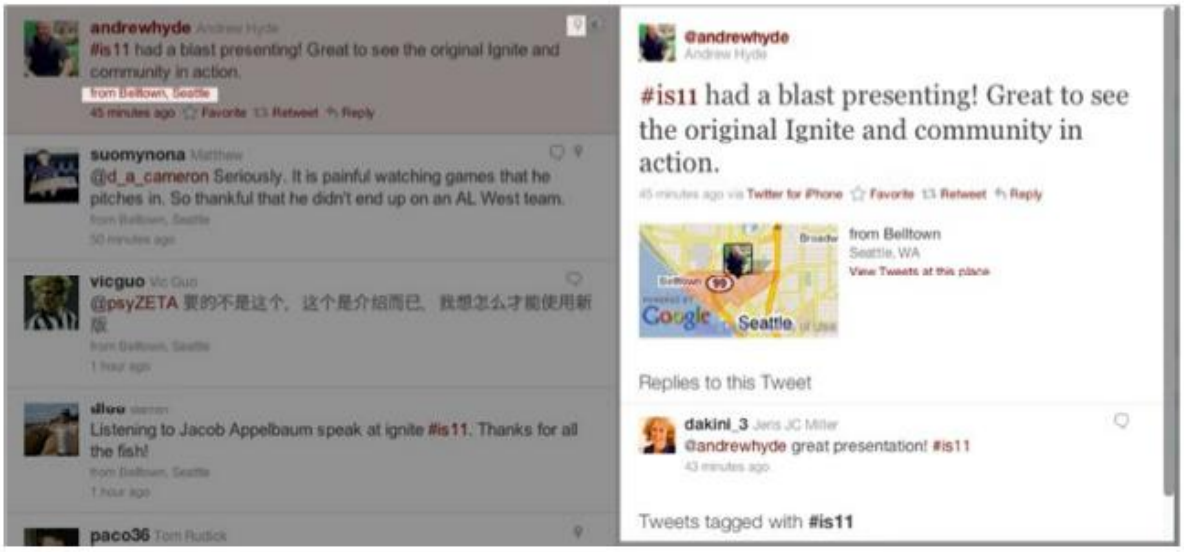

Figure 4: An example of geo-tagged tweet

The tweets from these cities can be used to achieve the following objectives.

- Model the spread of interests around the world. The data containing geo location Coordinates can be used to find out the content similarity between the tweets around different cities in the world to discover the keywords that are popular throughout the world. This information can also be used to target special interest groups in different cities using different campaigns. There have been several researches in this field of social information modeling based on locations. The authors in outline navigational and social aspects of such location based systems. Such an analysis can be used to analyze the timed information of different events, locations of different keywords as well as the rate of information flow.

- Predict future/current events. The tweets' data collected from different cities has been used to predict future and current events as well as the result of elections, popularity of movies etc. As well as prediction of disaster events.

3.3.2 Tweets about a topic

The next set of tweets that I collect are based on a set of keywords that describe a topic in real world. I collect tweets that contain the following two sets of keywords:

- Tweets about Apple Inc.16: apple, mac, macbook, macbookair, macbookpro, os x,osx, osxlion, ipod, ipodshuffle, ipodnano, ipodclassic, ipodtouch, itunes, iphone,iphone3, iphone3s, iphone4, iphone4s, iphone5, ios, ios4, ios5, ipad, ipad2, ipad3.

Figure 5: An example tweet about Apple Inc.

- Tweets about Manchester United17: manchesterunited, manchester united, manchester utd, man united, manutd, man utd, manu, mufc .

Figure 6: An example tweet about Manchester United

These set of tweets allow us to achieve different goals. These set of tweets again serve the goal of modeling the spread of user interests around the world as well as the popularity of these topics at different points in time. This can again be used to model the rate of information flow on the internet. The collection of tweets from these keywords can also help these organizations to target

different user groups in different places as well as try to obtain product reviews and popularity of soccer matches. This type of modeling has been done in the past with the goal of marketing for different companies.

3.3.3 Tweets from a group of users

Finally, I also collect tweets from a group of users on twitter. I collected this group of users by looking at the friends and followers of a central user18. I first collected the users that follow the central user and the users that are being followed by him. That is I collect users that have any of the two kinds of links with the central users. I then do the same for the users collected in the previous step. This means that I collect the followers and friends of a central user up to two hops in the social connection hierarchy. Our aim with the collection of these users is to analyze the meaningful connections between these groups of users and therefore, I excluded celebrities or other very popular users (users which have more than 1000 followers or follows more than 5000 other users) from our study as these would have many relationships outside a tightly connected community of users.

Figure 7 presents an overview of the collection system. The links show a directed relationship between users. A link from user ‘a’ to user ‘b’ means that ‘a’ follows ‘b’ on twitter. The being followed relation has not been shown in the figure as it is just the reverse of the following relationship between users. The blue links are the links that I follow to collect the users. The node with dark blue color in the centre represents the central user that I use as a starting point for our collection system. The red links are the links at which I stop collecting the users further. This allows us to limit our system to a limited number of users as compared to the large user base of twitter.

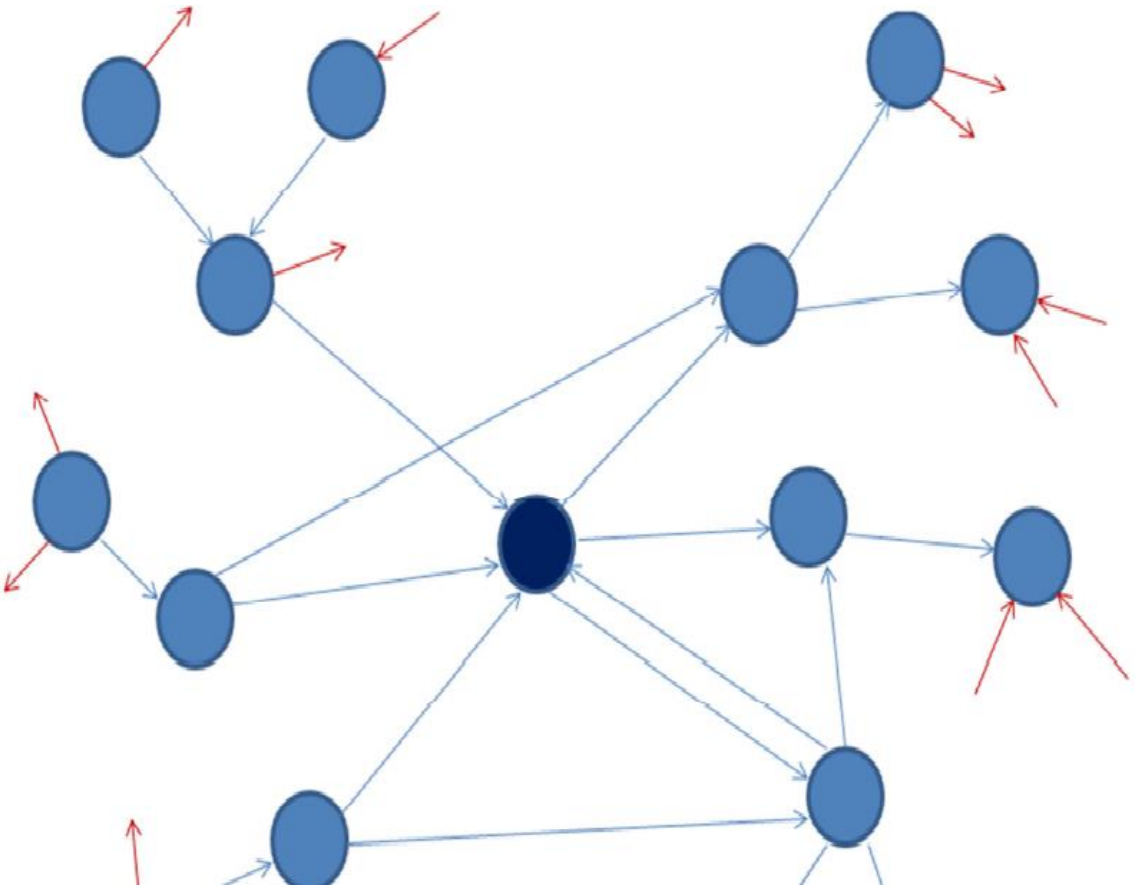

Figure 7: Overview of links I use to collect users

I collected all the public profile information of the users that belong to these groups. In addition to this, I also collect the tweets that these users post. The collection of this information allows us to model different types of social link and relationships between these users. Further in the report I will also present a detailed analysis of the social information that I obtain and how I use this information to detect the community structure in this group of users. The collection of this information allows us to achieve the following goals.

- Model relationships among users. As explained before, I can use the social connections between these users to model their relationships on twitter and also detect a community structure that corresponds to a tight cluster of these users on twitter.

- In addition to this, I can also model the common interest of these users using the content of their tweets. I also use several other meta-data contained in the tweets to cluster the users.

3.4 Visualizations

I now present a few visualizations and analysis on the data collected using the locations and keywords that allow us to draw certain simple inferences from the above tweet data.

3.4.1. Visualization of tweets collected by location

I begin by presenting a very simple analysis of how the geo-tagged tweets from a city can be used to identify some places of interest in the city. Without any loss of generality, let us present an example of visualizing tweets in London. These are the geo-tagged tweets collected for one week (16 Aug, 2011 to 22 Aug, 2011) from London as per the bounding box coordinates given earlier (in Section 2.3.1). I plot these tweets as small blue points on a map of London using Geo-Commons19.

Figure 8 contains the tweets for the topic 'Manchester United' in the specified time frame. By looking at the visualization results, I can infer that most of the tweets mentioning Manchester United come from in and around Europe. This can be because of the fact that Manchester United plays in the English Premier League and has its home ground in Manchester. In addition to this, I also find that there are a lot of tweets from countries whose players play for Manchester United. I also present a few such examples in the visualization where I show a tweet mentioning the player 'Nani' coming from Portugal and another tweet mentioning the player 'Anderson' from Brazil. In addition to these inferences, I also find that there are a lot of tweets from Indonesia and Malaysia that talk about Manchester United. This is because of the fact that Manchester United has invested a lot in these countries and is therefore very popular.

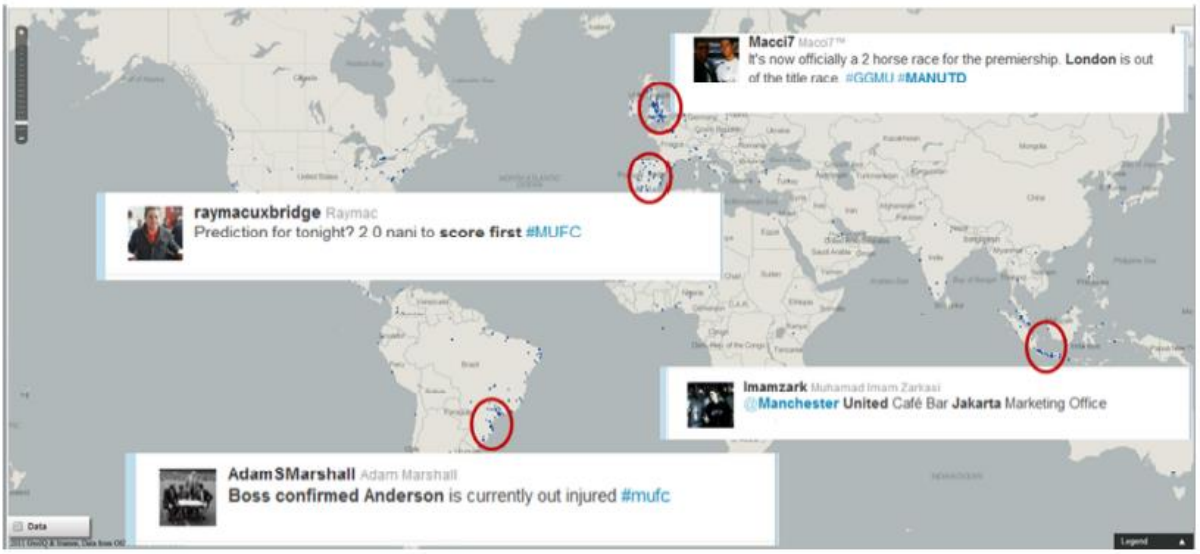

Figure 8: Tweets about the topic 'Manchester United'

Figure 9 on the other hand shows the geo-tagged tweets about the topic ‘Apple Inc.’. I find that as opposed to tweets about Manchester United which were mostly from Europe, Apple has a much larger popularity and tweets about Apple come mostly from North America and Europe. This can be explained as the popularity and usage of products of Apple in these regions.

When I compare the results of Apple and Manchester United, I can see that Apple is more popular than Manchester United as the volume of tweets for Apple is much larger than Manchester United. E.g. For the above setup of two weeks, I obtained more than 32,000 geo-tagged tweets for Apple as opposed to only 1,400 geo-tagged tweets for Manchester United. Another inference that I can draw from the above visualizations is that interests about Apple are spread over the world whereas for Manchester United, the interests are restricted mostly to Europe and few countries in Asia.

Figure 9: Tweets about the topic 'Apple'

3.5. Community Detection

3.5.1. Background

Community detection/clustering is the process of taking collections of objects such as tweets, location similarity and organizing them into groups based on their similarity. The organization into groups should be such that similar objects belong to the same cluster whereas there is little or no similarity between objects that belong to different clusters. The main elements of the problem themselves, i. e. the concepts of community and partition, are not rigorously defined, and require some degree of arbitrariness and/or common sense. It is important to stress that the identification of structural clusters is possible only if graphs are sparse, i. e. if the number of edges m is of the order of the number of nodes n of the graph. If m >> n, the distribution of edges among the nodes is too homogeneous for communities to make sense.

Before looking into the clustering algorithms for graphs, let us first discuss about the notion of communities in social networks. There is not one globally accepted definition for communities in social networks. But, from intuition, one can say that the communities should have an important

property that the nodes inside a community should have more connections among them rather than between nodes from different communities. Communities are the parts of graph with few ties with the rest of the system. There have been several researches in the field of clustering in the past. The clustering algorithms can be grouped into two major classes:

3.5.2. Hierarchical Clustering

In general, since very little is known about the community and its structure in the network, it is difficult to estimate the number of clusters beforehand. In such cases, one needs to apply specific algorithms on graphs in order to determine the community structure in graphs. Often, it requires making certain assumptions about the number and size of clusters in the graph. On the other hand, there can be certain hierarchical structure in the graph which can be exploited in order to detect communities in graph. The starting point of hierarchical clustering algorithms is a measure of similarity between the nodes in the graph. The hierarchical algorithms are further divided into the following different classes:

a. Agglomerative algorithms: These algorithms iteratively merge two different clusters if their similarity is sufficiently large. It is a bottom up process which starts with each node as a different cluster and then merges the clusters based on the similarity between clusters. Since clusters are merged based on their mutual similarity, it is essential to determine a measure that estimates how similar clusters are. This involves some arbitrariness and several prescriptions exist. In single linkage clustering, the similarity between two groups is the minimum element xij , with i in one group and j in the other. On the contrary, the maximum element xij for vertices of different groups is used in the procedure of complete linkage clustering. In average linkage clustering one has to compute the average of the xij.

b. Divisive algorithms: These algorithms iteratively split a cluster by removing edges connecting vertices with low similarity. Newman-Girvan [5] is an algorithm that is based on divisive clustering and has been used extensively in the past for community detection. The

algorithm proceeds by finding the edges with maximum edge between ness and removing such edges.

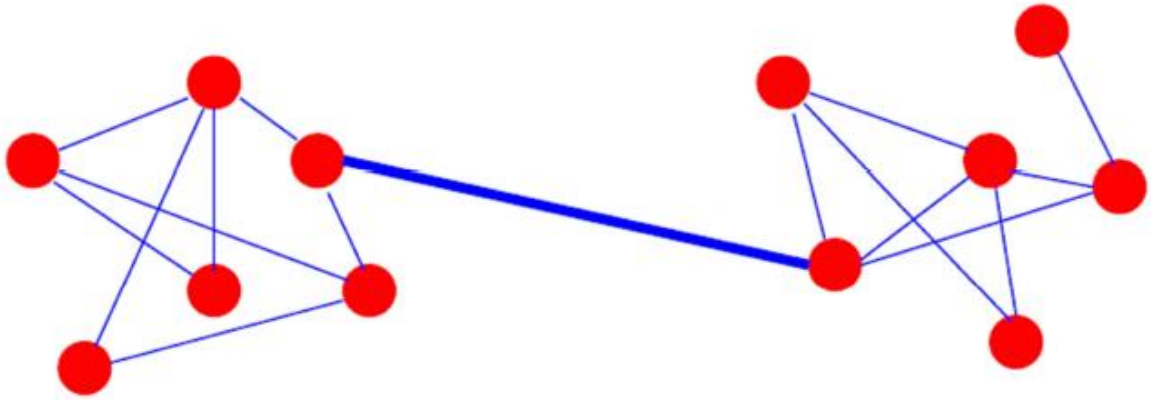

Figure 10: Edges connecting two groups has highest edge between ness

Generally the clustering algorithm imposes certain restrictions on the number of clusters or quality criterion (e.g. modularity) to find the correct distribution of clusters. The results of a hierarchical clustering algorithm are generally represented in form of dendrogram.

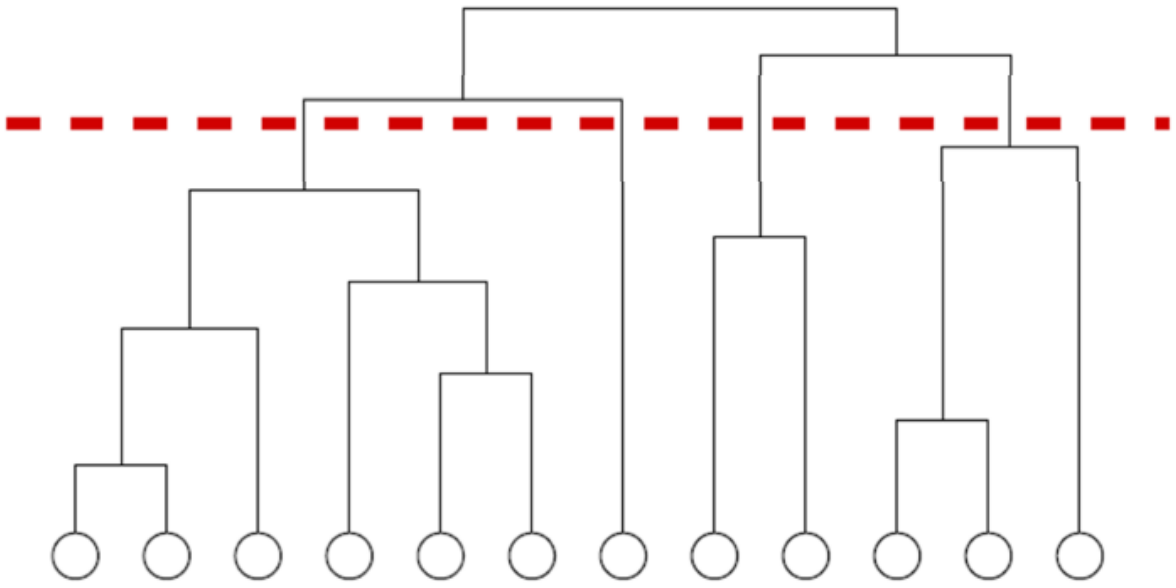

Figure 11: A dendrogram or hierarchical tree

CHAPTER 4

Results and Analysis

In this section I will present an analysis on twitter data set collected for three # tag keywords, Samsung, Sony, LG. I have checked the popularity of those words on twitter, so that it can be possible to make a decision making system on behalf of the popularity of the particular keyword.

By comparing with positive negative words set, I identified whether the twitter's user is saying positive or negative about that particular keyword. If a tweet contains one positive and one negative word then the tweets score will be consider as zero as +1 point is for positive word and -1 is for negative word. If more positive/negative words are used in a tweet sentence then the score will be accordingly.

As we can see from the Figure 12, samsung score chart, around 2000 tweets are collected from the twitter which was having samsung keyword as a hash tag. In those 2000 tweets around 1800 tweets are having score 0, around 200 tweets having 1 score and around 10 tweets are having score 2.

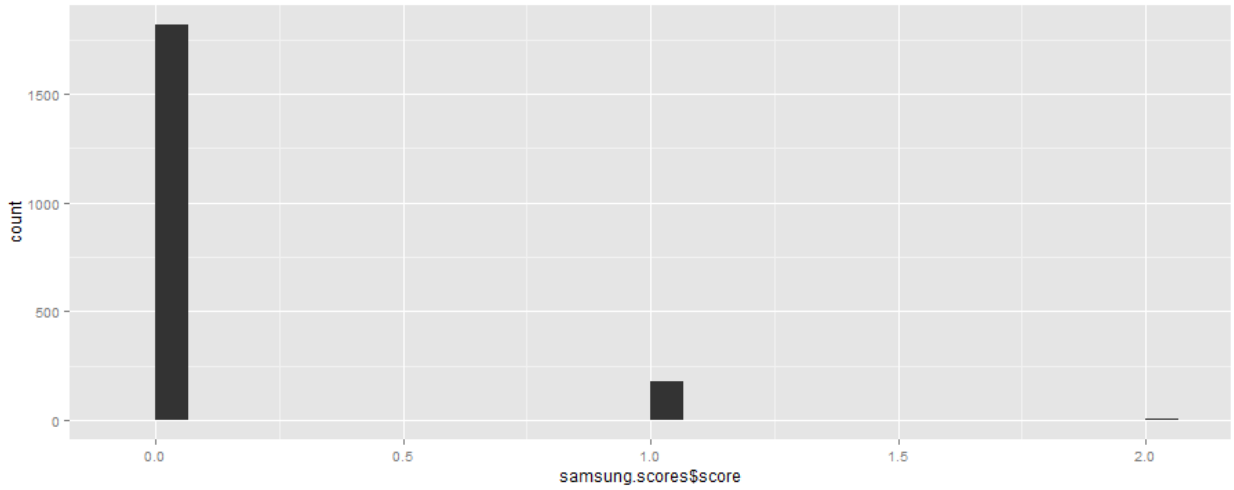

Figure 12 Samsung Score Chart

Same type of analysis can be done for sony hash tag keyword. As we can see from the Figure 13, sony score chart, around 1100 tweets are collected from the twitters which were having sony keyword as a hash tag. In those 1100 tweets around 1000 tweets are having score 0, around 100 tweets having 1 score and around 10 tweets are having score 2, and 5 to 6 tweets are having score 3 or 4.

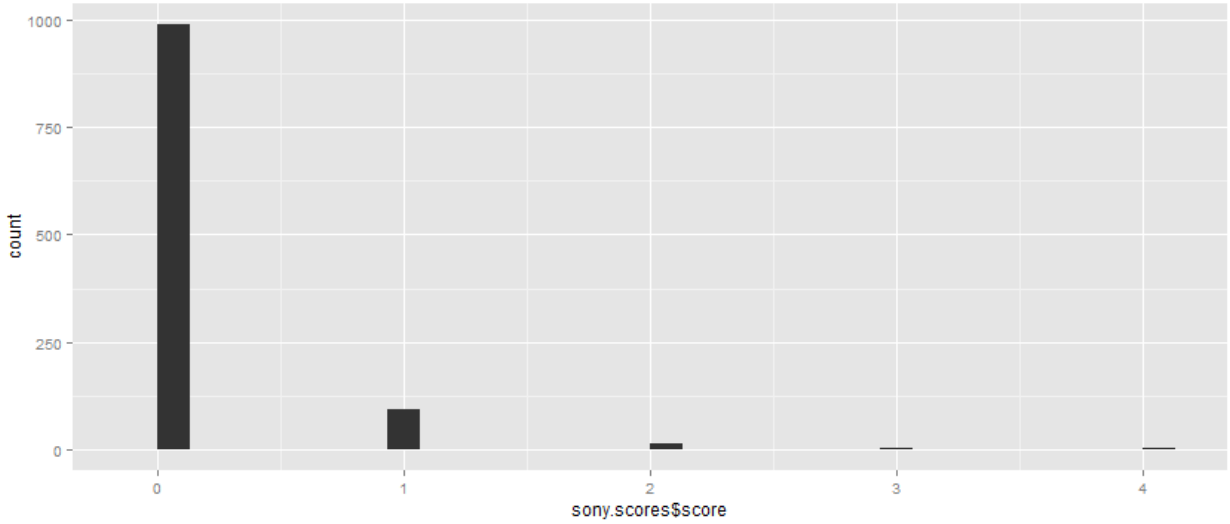

Figure 13 Sony Score Chart

Same type of analysis can be done for LG hash tag keyword. As we can see from the Figure 14, LG score chart, around 1200 tweets are collected from the twitters which were having log keyword as a hash tag. In those 1200 tweets around 1100 tweets are having score 0, around 50 tweets having 1 score and around 5 tweets are having score 2, and 5 tweets are having score 3.

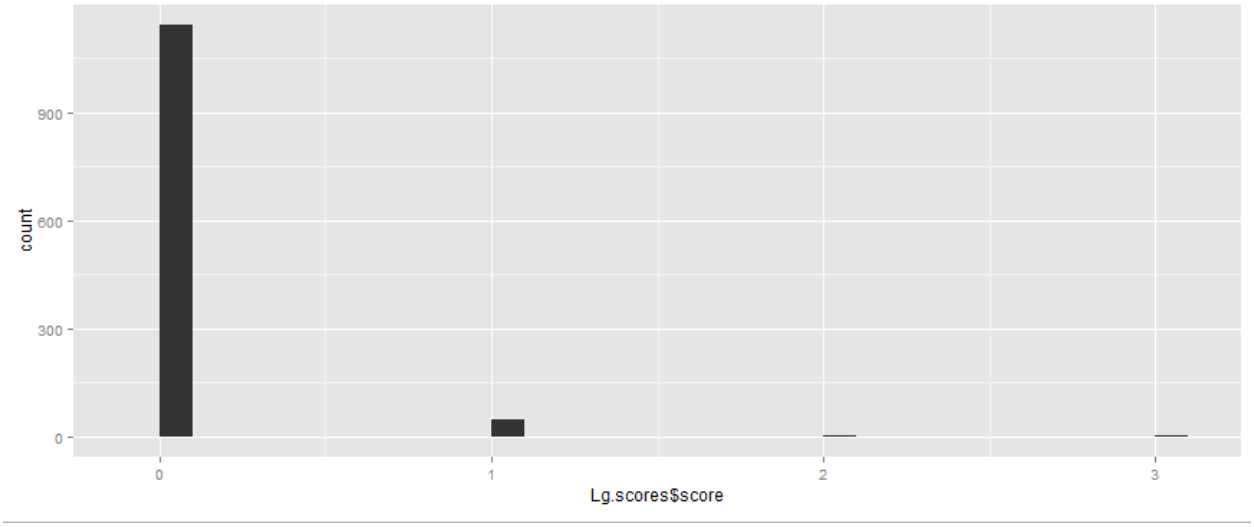

Figure 14 LG Score Chart

We can compare all the above 3 tweets (Samsung, Sony, LG) chart together to get the better understanding by visualization. In Figure 15 we can see the comparison on the same scale.

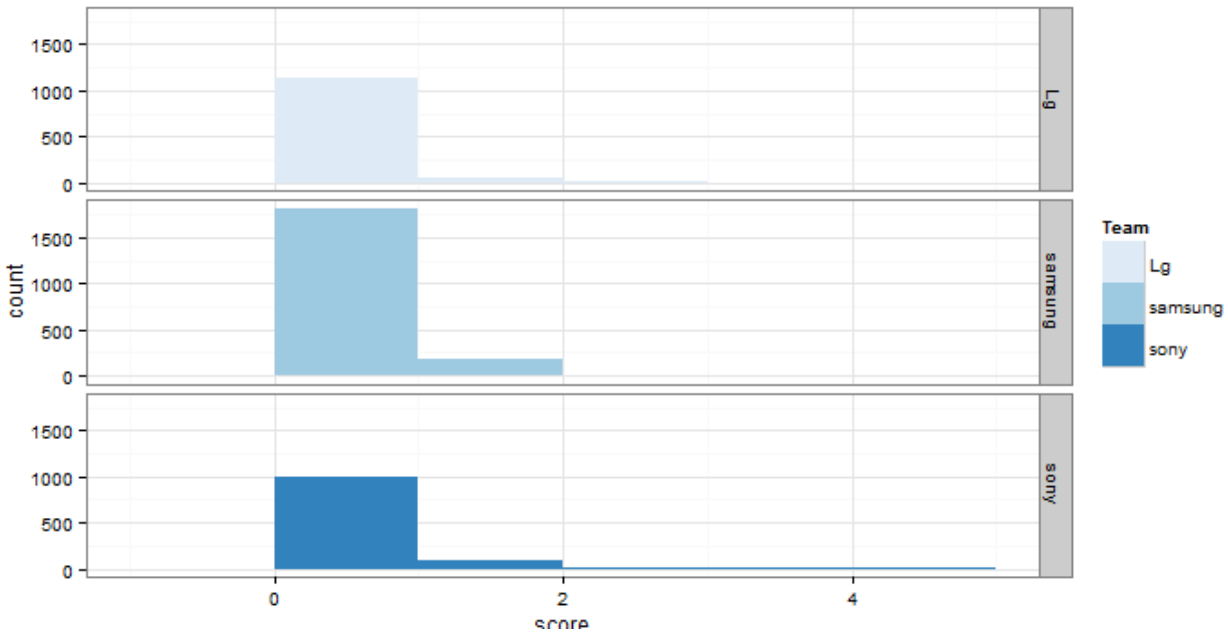

Figure 15 Comparison Chart

Hash Tag Keyword	Score 0	Score 1	Score 2	Score 3
Samsung	1800	200	10	
Sony	1000	100	10	5
LG	1100	50	5	5

Table 1 Comparison of the tweets data

In the below figure 16, samsung tweets are represented by the graph. These graphs are very useful to identify the node from where the more tweets are coming. If some user doing negative marketing for any company it can be identified and culprit can be catched very easily. And we can use this graph in much future analysis.

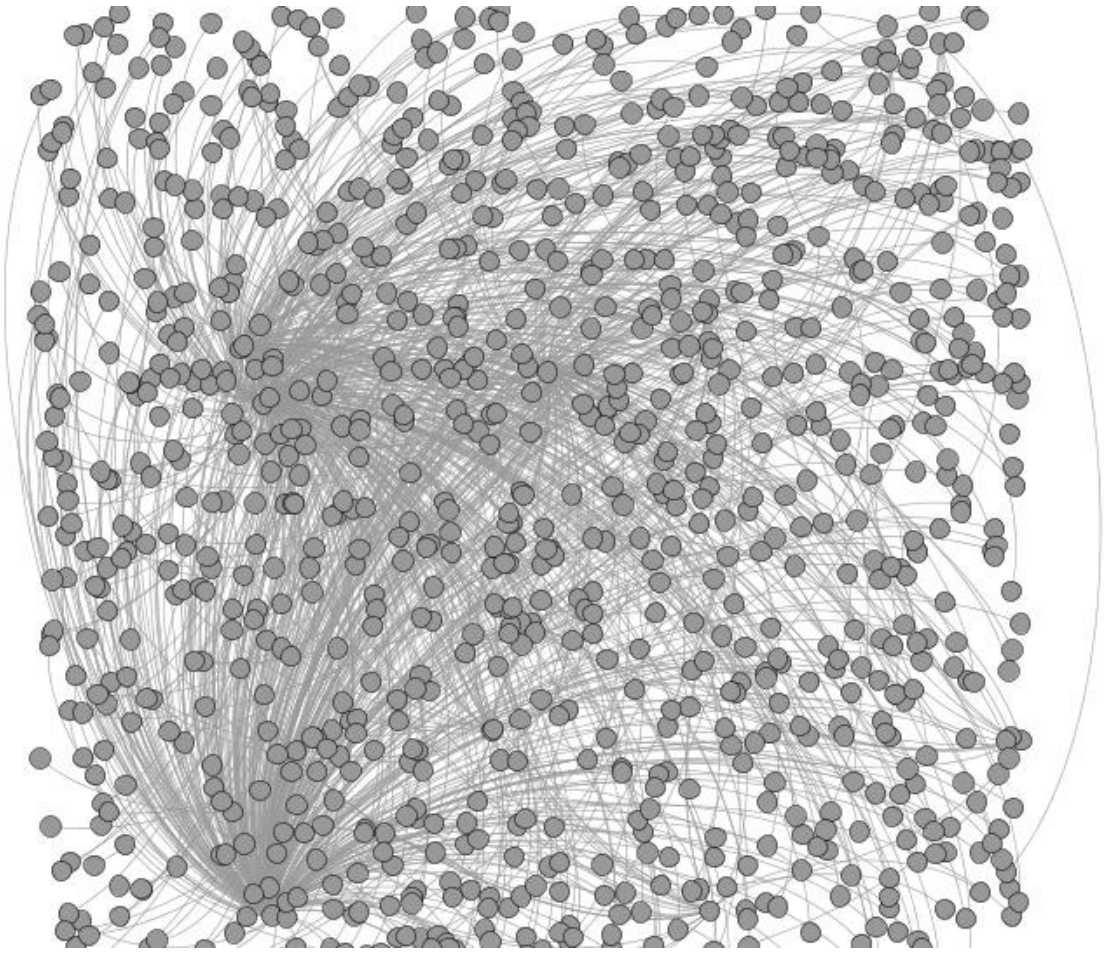

Figure 16 Graph Representations of Samsung Tweets

References

[1] Normaslina Jamil, Arifah Che Alhadi, Shahrul Azman Noah : A Collaborative Names Recommendation in the Twitter Environment based on Location, International Conference on Semantic Technology and Information Retrieval,2011

[2] Zhiheng Xu, Rong Lu ,Liang Xiang ,Qing Yang: *Discovering User Interest on Twitter with a Modified Author-Topic Model*, IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology.2011

[3] Elenna R. Dugundji, Ate Poorthuis, and Michiel van Meeteren: Modeling user behavior in adoption and diffusion of Twitter clients, IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing, 2011

[4] Renxian Zhang, Wenjie Li, Dehong Gao, and You Ouyang: Automatic Twitter Topic Summarization With Speech Acts, IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 21, NO. 3, MARCH 2013

[5] Mengjiao Wang and Donn Morrison and Conor Hayes: Early and late fusion methods for the automatic creation of Twitter lists, IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2012

[6] Erick Stattner, Social Network Analysis in Epidemiology: Current Trends and Perspectives IEEE International Conference on Social Computing, 2012

[7]Hui Wang,Lin Deng: A VARIANT EPIDEMIC PROPOGATION MODEL SUITABLE FOR RUMOR SPREADING IN ONLINE SOCIAL NETWORK, Proceedings of the 2012 International Conference on Machine Learning and Cybernetics, Xian, 15-17 July, 2012

[8] Ryan Compton, Lalindra De Silva: Detecting future social unrest in unprocessed Twitter Data

[9] Erica Rodrigues, Renato Assunc¸ ˜ao,: Uncovering the location of *Twitter* users, Brazilian Conference on Intelligent Systems, 2013

[10] D. Simmie, M. G. Vigliotti, C. Hankin: Ranking Twitter Influence by Combining Network Centrality and Influence Observables in an Evolutionary Model, International Conference on Signal-Image Technology & Internet-Based Systems, 2013

Appendix 1

Source Code

```
##############################################

# Chunk - 1 - Authenticate with twitter API

##############################################

install.packeges("twitteR")

install.packeges("plyr")

install.packeges("stringr")

install.packeges("ggplot2")

library(twitteR)

library(ROAuth)

library(plyr)

library(stringr)

library(ggplot2)

## Windows users need to get this file

download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem")

requestURL <- "https://api.twitter.com/oauth/request_token"

accessURL = "https://api.twitter.com/oauth/access_token"

authURL = "http://api.twitter.com/oauth/authorize"
```

```
consumerKey = "Uj6StYePOjiVONi9SyaPQA"

consumerSecret = "QzQ7MvIF1FZw4cWlHavOplcPlMejAVl789sx0CytRcI"

Cred <- OAuthFactory$new(consumerKey=consumerKey,

                         consumerSecret=consumerSecret,

                         requestURL=requestURL,

                         accessURL=accessURL,

                         authURL=authURL)

Cred$handshake(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl") )

0742171

save(Cred, file="twitter authentication.Rdata")

registerTwitterOAuth(Cred)

## Future use

load("twitter authentication.Rdata")

registerTwitterOAuth(Cred)

#################################################################

# Chunk  - 2 - Twitter Scrape  #Rangers #Athletics #MLB

#################################################################

samsung.list <- searchTwitter('#samsung', n=2000, cainfo="cacert.pem")

samsung.df = twListToDF(samsung.list)

write.csv(samsung.df, file='C:/temp/samsungTweets.csv', row.names=F)

sony.list <- searchTwitter('#sony', n=2000, cainfo="cacert.pem")
```

```
sony.df = twListToDF(sony.list)

write.csv(sony.df, file='C:/temp/sonyTweets.csv', row.names=F)

Lg.list <- searchTwitter('#Lg', n=2000, cainfo="cacert.pem")

Lg.df = twListToDF(Lg.list)

write.csv(Lg.df, file='C:/temp/LgTweets.csv', row.names=F)

################################

#Chunk -3- Sentiment Function

################################

library (plyr)

library (stringr)

score.sentiment = function(sentences, pos.words, neg.words, .progress='none')

{

  require(plyr)

  require(stringr)

  # we got a vector of sentences. plyr will handle a list

  # or a vector as an "l" for us

  # we want a simple array ("a") of scores back, so we use

  # "l" + "a" + "ply" = "laply":

  scores = laply(sentences, function(sentence, pos.words, neg.words) {

    # clean up sentences with R's regex-driven global substitute, gsub():

    sentence = gsub('[[:punct:]]', '', sentence)
```

```
  sentence = gsub('[[:cntrl:]]', '', sentence)


  sentence = gsub('\\d+', '', sentence)
  # and convert to lower case:
  sentence = tolower(sentence)
  # split into words. str_split is in the stringr package
  word.list = str_split(sentence, '\\s+')
  # sometimes a list() is one level of hierarchy too much
  words = unlist(word.list)
  # compare our words to the dictionaries of positive & negative terms
  pos.matches = match(words, pos.words)
  neg.matches = match(words, neg.words)
  # match() returns the position of the matched term or NA
  # we just want a TRUE/FALSE:
  pos.matches = !is.na(pos.matches)
  neg.matches = !is.na(neg.matches)
  # and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
  #score = sum(pos.matches) - sum(neg.matches)
#score = sum(pos.matches)
score = sum(neg.matches)
##
```

```
    return(score)


  }, pos.words, neg.words, .progress=.progress )
  scores.df = data.frame(score=scores, text=sentences)
  return(scores.df)
}


##############################################
#Chunk - 4 - Scoring Tweets & Adding a column
##############################################
#Load sentiment word lists
hu.liu.pos = scan('C:/temp/positive-words.txt', what='character', comment.char=';')
hu.liu.neg = scan('C:/temp/negative-words.txt', what='character', comment.char=';')
#Add words to list
pos.words = c(hu.liu.pos, 'upgrade')
neg.words = c(hu.liu.neg, 'wtf', 'wait','waiting', 'epicfail', 'mechanical')
#Import 3 csv
Datasetsamsung <- read.csv("C:/temp/samsungTweets.csv")
Datasetsamsung$text<-as.factor(Datasetsamsung$text)
Datasetsony <- read.csv("C:/temp/sonyTweets.csv")
Datasetsony$text<-as.factor(Datasetsony$text)
```

```
DatasetLg <- read.csv("C:/temp/LgTweets.csv")

DatasetLg$text<-as.factor(DatasetLg$text)


#Score all tweets

samsung.scores = score.sentiment(Datasetsamsung$text, pos.words,neg.words, .progress='text')

sony.scores = score.sentiment(Datasetsony$text, pos.words,neg.words, .progress='text')

Lg.scores = score.sentiment(DatasetLg$text, pos.words,neg.words, .progress='text')

path<-"C:/temp/"

write.csv(samsung.scores,file=paste(path,"samsungScores.csv",sep=""),row.names=TRUE)

write.csv(sony.scores,file=paste(path," sonyScores.csv",sep=""),row.names=TRUE)

write.csv(Lg.scores,file=paste(path,"LgScores.csv",sep=""),row.names=TRUE)

samsung.scores$Team = 'samsung'

sony.scores$Team = 'sony'

Lg.scores$Team = 'Lg'


##############################

#Chunk -5- Visualizing

##############################

hist(samsung.scores$score)

qplot(samsung.scores$score)

hist(sony.scores$score)
```

```
qplot(sony.scores$score)


hist(Lg.scores$score)

qplot(Lg.scores$score)


##################################

#Chunk -6- Comparing 3 data sets

##################################

all.scores = rbind(samsung.scores, sony.scores, Lg.scores)

ggplot(data=all.scores) + # ggplot works on data.frames, always

  geom_bar(mapping=aes(x=score, fill=Team), binwidth=1) +

  facet_grid(Team~.) + # make a separate plot for each hashtag

  theme_bw() + scale_fill_brewer() # plain display, nicer colors

##################################

#Chunk -7- Graph representation

##################################

library(igraph);

# Read Twapperkeeper CSV file

#tweets <- read.csv('C:/temp/tweets.csv', head=T, sep="|", quote="", fileEncoding="UTF-8");

#print(paste("Read ", length(tweets$text), " tweets.", sep=""));

tweets <- read.csv("C:/temp/samsungtweets.csv")
```

```
tweets$text<-as.factor(tweets$text)

# Get @-messages, senders, receivers

ats <- grep("^\\.?@[a-z0-9_]{1,15}", tolower(tweets$text), perl=T, value=T);

at.sender <- tolower(as.character(tweets$screenName[grep("^\\.?@[a-z0-9_]{1,15}",
tolower(tweets$text), perl=T)]));

at.receiver <- gsub("^\\.?@([a-z0-9_]{1,15})[^a-z0-9_]+.*$", "\\1", ats, perl=T);

print(paste(length(ats), " @-messages from ", length(unique(at.sender)), " senders and ",
length(unique(at.receiver)), " receivers.", sep=""));


# Get RTs, senders, receivers

rts <- grep("^rt @[a-z0-9_]{1,15}", tolower(tweets$text), perl=T, value=T);

rt.sender <- tolower(as.character(tweets$screenName[grep("^rt @[a-z0-9_]{1,15}",
tolower(tweets$text), perl=T)]));

rt.receiver <- gsub("^rt @([a-z0-9_]{1,15})[^a-z0-9_]+.*$", "\\1", rts, perl=T);

print(paste(length(rts), " RTs from ", length(unique(rt.sender)), " senders and ",
length(unique(rt.receiver)), " receivers.", sep=""));


# This is necessary to avoid problems with empty entries, usually caused by encoding issues in
the source files

at.sender[at.sender==""] <- "<NA>";

at.receiver[at.receiver==""] <- "<NA>";

rt.sender[rt.sender==""] <- "<NA>";

rt.receiver[rt.receiver==""] <- "<NA>";
```

```
# Create a data frame from the sender-receiver information
ats.df <- data.frame(at.sender, at.receiver);
rts.df <- data.frame(rt.sender, rt.receiver);

# Transform data frame into a graph
ats.g <- graph.data.frame(ats.df, directed=T);
rts.g <- graph.data.frame(rts.df, directed=T);

# Write sender -> receiver information to a GraphML file
print("Write sender -> receiver table to GraphML file...");
write.graph(ats.g, 'C:/temp/ats_samsung.graphml', format="graphml");
write.graph(rts.g, 'C:/temp/rts_samsung.graphml', format="graphml");
```