University of Cincinnati

Date: 11/8/2011

I, Sathvik Nerupalli, hereby submit this original work as part of the requirements for the degree of Master of Science in Computer Science.

It is entitled:
Personalized User Trending Topics

Student's name: Sathvik Nerupalli

This work and its defense approved by:

Committee chair: Fred Annexstein, PhD
Committee member: Kenneth Berman, PhD
Committee member: Anca Ralescu, PhD
Personalized User Trending Topics

A thesis submitted to the
Graduate School
of the University of Cincinnati
In partial fulfillment of the requirements for the degree of

Master of Science

In Computer Science
at School of Computing Sciences and Informatics
of the College of Engineering and Applied Science

By
Sathvik Nerusupalli
Bachelor of Technology, Computer Science and Engineering,
Sreenidhi Institute of Science and Technology, India

Committee Chair: Dr. Fred S. Annexstein, Ph.D.
Abstract

Online Social networks have a cornucopia of interesting information waiting to be tapped and understood. Trending topics is one of the easier ways to understand the inclination of information that is being published in social networks. Trending topics are generic and might not be of interest to a certain individual on every occasion. Personalized user trends are the trending topics that occur in the neighborhood of a user. Further, in this thesis, we discuss techniques to find personalized user trends by using a new process in stop word removal that is effective in dynamic environment like Twitter and in identifying features that affect trending topics in a user’s neighborhood.
Acknowledgements

I am indebted to my advisor, Dr. Fred Annexstein, whose encouragement, inspiration and perpetual enthusiasm in the subject drove me to do this thesis and complete it successfully. I also extend my thanks to my other committee members, Dr. Kenneth Berman and Dr. Anca Ralescu, who were always there through my Master’s degree in providing necessary course work and foundation for this work.

I also thank my friends who shared my thoughts, brainstormed with me, and helped me think differently and guided me through the whole process. Particularly, I would like to thank Vindhya, for her never ending patience and love, Praneeth, for his inquisitive questions that helped me clear my thoughts, Purushotham, for his perpetual willingness to help, Naren, for his strong support, Chaitanya, for always keeping me abreast with the latest technology and all my friends at University of Cincinnati.

Finally, I wish to thank University of Cincinnati, which provided me with great infrastructure, resources and people!
# Table of Contents

1. __Introduction__ .......................................................................................................................... 2
   1.1. The problem statement ........................................................................................................... 3
   1.2. Personalized Trending Topics ............................................................................................... 3
   1.3. Motivation .............................................................................................................................. 4
      1.3.1. Analyzing personal usage ............................................................................................. 4
      1.3.2. Analyzing commercial usage ....................................................................................... 5
   1.4. Achievements ......................................................................................................................... 6
   1.5. Constraints and Assumptions ............................................................................................... 7

2. __Background__ ............................................................................................................................ 8
   2.1. Twitter .................................................................................................................................. 8
      2.1.1. Overview of Twitter ....................................................................................................... 8
   2.2. Concepts and terms ............................................................................................................... 10
      2.2.1. Trending Topic .............................................................................................................. 10
      2.2.2. Circles ............................................................................................................................ 11
   2.3. Related Work ....................................................................................................................... 12

3. __Implementation__ ...................................................................................................................... 13
   3.1. Technology Used .................................................................................................................. 13
      3.1.1. Coding: Java .................................................................................................................. 13
      3.1.2. Access to Twitter API: Twitter4j .................................................................................. 13
      3.1.3. Backend: MySQL .......................................................................................................... 13
      3.1.4. IDE: Net beans ............................................................................................................... 14
   3.2. Flow Charts ......................................................................................................................... 14
      3.2.1. Pre-requisites ................................................................................................................. 14
      3.2.2. Stage I – Initial Data Collection .................................................................................... 14
      3.2.3. Stage IIa - Word Extraction .......................................................................................... 15
      3.2.4. Stage IIb - Feature Extraction ....................................................................................... 16
      3.2.5. Stage IIc – Trending Topics ......................................................................................... 17
      3.2.6. Stage III – Real-time data collection ............................................................................. 17
   3.3. Significant Features: Tweet Level Features .......................................................................... 18
      3.3.1. Time Stamp and Epoch Time ........................................................................................ 18
      3.3.2. Mentions Score ............................................................................................................. 21
      3.3.3. ReTweet Score ............................................................................................................. 22
      3.3.4. Favorite Score .............................................................................................................. 23
Appendix A

Bibliography

Future Scope

Experimental Setup and Results

Trending Topics

All-Time Trending Topics

Conclusion

Bibliography

Appendix A

Stage I – Initial Data Collection

Get users timeline
8.2.2. Get follower Ids........................................................................................................................................43
8.2.3. Get followers timelines............................................................................................................................43
8.3. Stage IIa - Word Extraction..........................................................................................................................43
  8.3.1. Lexical Analysis........................................................................................................................................43
8.4. Stage IIb - Feature Extraction.....................................................................................................................44
  8.4.1. Tweet level features................................................................................................................................45
  8.4.2. Word level features................................................................................................................................45
8.5. Stage IIc – Trending Topics..........................................................................................................................45
8.6. Stage III – Real-time data collection............................................................................................................45
8.7. Exception Handling......................................................................................................................................46
8.8. Design Patterns: Singleton Pattern.............................................................................................................46
8.9. Database Organization.................................................................................................................................47
  8.9.1. Tables for Stage I and Stage III .............................................................................................................47
  8.9.2. Tables for Stage II................................................................................................................................50
8.10. Structure of a tweet......................................................................................................................................51
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Twitter Profile Page</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Authentication</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>Flowchart of the Stage I – Data Collection</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>Flowchart of Stage Iia - Word Extraction</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>Flowchart of Stage Iib - Feature Extraction</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>Stage Iic - Trending Topics</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>Stage III - Real-time Data Collection</td>
<td>18</td>
</tr>
<tr>
<td>8</td>
<td>Epoch Time: Scenario 1</td>
<td>19</td>
</tr>
<tr>
<td>9</td>
<td>Epoch Time: Scenario 2</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>Epoch Time</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>Graph of Top-35 words</td>
<td>29</td>
</tr>
<tr>
<td>12</td>
<td>Singleton Pattern</td>
<td>47</td>
</tr>
<tr>
<td>13</td>
<td>Tweets Database Table</td>
<td>48</td>
</tr>
<tr>
<td>14</td>
<td>Users Database Table</td>
<td>48</td>
</tr>
<tr>
<td>15</td>
<td>Followers Database Table</td>
<td>49</td>
</tr>
<tr>
<td>16</td>
<td>HashTag Database Table</td>
<td>49</td>
</tr>
<tr>
<td>17</td>
<td>URL Database Table</td>
<td>50</td>
</tr>
<tr>
<td>18</td>
<td>&lt;username&gt; Database Table</td>
<td>50</td>
</tr>
<tr>
<td>19</td>
<td>Stop words/ Expelled words Database Table</td>
<td>51</td>
</tr>
</tbody>
</table>
1. Introduction

Most of the human life is spent interacting with other people and in developing an ever growing complex and labyrinthine social network. The growth of internet in 1990’s has spawned the first ever online social network called the SixDegrees.com [1]. Online social networks were quickly embraced by people and have grown to be the most visited sites on the internet. In February 2009, total social network usage exceeded web-based mail usage for the first time [2]. Data from social networks have fundamentally changed the way we obtain, interpret and understand information. Information publicly available from social networks is huge and contains many details. Information provided in online social networks typically contain personally identifiable information which include their profile, preferences and connections.

Social networks started affecting big companies in various ways from marketing campaigns to hiring decisions on employees. Some companies including UK secret intelligence service announced their interest in social network research [2]. Marketing research costs can be reduced by utilizing the open data available in online social networks. The ways we can derive for utilization of online social networks are endless.

There are many online social networks that came into limelight in the past few years. MySpace, Orkut, Facebook, Twitter and Google plus are some of the well-known names in online social networks. Every online social network distinguishes itself by exposing various levels of information about users. For this thesis, we picked Twitter for testing our techniques. Twitter is one of the most approachable social networks with well exposed APIs to access their data. In the following chapters we will explain the basic terminology and build up the required knowledge base for the problem. Firstly, we discuss about the thesis statement, motivation of the thesis and
the reason why this problem is important. Secondly, we introduce various terminologies and concepts on which we build the techniques. After understanding the necessary terminology of Twitter, we discuss the existing literature. Next part explains the techniques we used to solve the problem and achieve the desired results. In the last part, we get into the results obtained through our experiments and conclude with the possibilities of extension of this work into various fields.

1.1. The problem statement

The thesis statement we worked on is as follows:

“Identifying personalized trending topics is important in applications such as online marketing, business analysis, etc. In this thesis, we develop a method to analyze tweets and identify personal trending topics by removing stop words and ranking the words in tweets based on the influential features. We successfully derived a method to identify personalized trending topics”

Our main contributions in this thesis are

- Developing an architecture to effectively receive the high tweet inflow
- Identify the main features that affect a trending topic to be interesting at a user level and also at a global level
- Identifying a set of techniques that can extract the actual trending topics at a personalized user level

1.2. Personalized Trending Topics

Trending topics we just mentioned actually mean global trending topics or the trending topics when the whole data is compared. Global trending topics sometimes might not mean anything to
user. User trending topics or personalized trending topics are the set of topics that are trending in a specific neighborhood of a user. These are important for personalizing and identifying user specific information. If we want to know what a user is interested in and what he might like, we need to know about user specific trending topics. There are many research topics based on classifying users: like news oriented, politics oriented, celebrity oriented etc. all these try to generalize user interests based on the category. Generalizing user interests into any category will not yield topics of interest rather believing in each user to have their own unique identity will actually project each user’s actual category of trending topics. Every user will have different set of ideas and tweets that he would like to know and tweet about, hence we believe user trending topics are the way of approaching user interest.

For example, consider the following scenario. A Marketing professional/ business analyst wants to entice a twitter user. He wants to know what a user likes and he needs to understand the user’s tastes in various aspects. This particular information forms an important weapon or tool that industry wants. They want the particular interests of specific users. This can only be obtained by analyzing a user’s personalized data in their neighborhood.

1.3. Motivation

1.3.1. Analyzing personal usage

Online social networks are web-based portals created by users to share information. Recently, with the amount of popularity gained by online social networks information overflow has become a commonplace scenario. Even with Twitter, with as little as 140 character limit, the data flow is huge.
To understand the dynamism in a social network, we need a summary of the information that is shared. There is a necessity to know what is happening around a user in his social circle to keep up-to-date with his friends. If we know the current trending topics in the pack of friends we have, it would be easy and interesting to go through them quickly and stay informed. In this thesis, we identify these trending topics at a personalized level.

Hiring Managers and Human Resource people of major companies have officially declared that online social network is one of their many criteria that they look at while hiring a person [3]. Many of the checks they perform include posting appropriate content, evaluating their content profile based on the information they post in online social networks.

1.3.2. Analyzing commercial usage

The architecture developed through this thesis is scalable and has scope to extend to bigger neighborhood or circles. This makes the implementation flexible and to integrate easily within an organization. Interests and trending topics received from this application can be used to analyze the information flow in a group and understand what kind of news makes it to the most number of people.

Online Marketing is expected to be one of the most enthusiastic users for this idea. Firstly, Online Surveying is a technique that most companies follow to gain information and feedback from their users. This is a costly and time consuming process. Most of the users will not be interested in spending the time necessary for online surveys. Online social networks provide information that contains almost all the information necessary for completion of an online survey. Online surveys would further need to be aggregated, processed and ranked accordingly
to identify the areas of interest. Using personalized trends, this process becomes highly refined. If companies are to target a specific set of users then they will be able to find the most trending topics in that group. These topics can be then classified according to the requirement of the companies to build online surveys of different categories of people.

Recently, companies are also employing newer techniques to impress targeted users who have a huge footprint in the follower numbers. If a company can impress such a user and make him tweet about their product, the company not only gets free ad base but also develops a trusted base of users. The firsthand information given by peers in twitter creates a huge impact on the people following them.

Online social networks started attracting companies’ right from their debut in the online arena. Online social networks have fundamentally transformed the amount of knowledge they can gain about users. Twitter is one of the best free resource of data about users from various backgrounds. A company would be interested in knowing the kind of data that a group of people might be interested in knowing. For this, they need to have an aggregated data of what is going on in a group. Our thesis provides a starting point for such research. We provide them with the trending topics of a group. These trending topics signify the summary of the discussions going on in the group.

1.4. Achievements

In this thesis, we propose a new idea of personalized trending topics. As of the date of this thesis, there are no actual implementations of this process or idea in any major online social networks.
Though, there are ways to observe tweets of friends and information of your groups, having trends on a local level is comparatively new.

Secondly, we propose a way to solve the problem of identifying personalized trending topics. The approach we take is very simplistic and does not use any advanced techniques and will be acting as a primer towards further work in this aspect. The work here is broadly divided into two major sections other than the data collection and other miscellaneous work section. They are

1. Stop word identification and removal
2. Significant feature identification

Lastly, we isolate the personalized trending topics from the words occurring in the tweets after eliminating the possible stop words and applying filters based on the features that make the trending topics differ from the casual words.

1.5. Constraints and Assumptions

In this thesis we encountered some problems that constrained us. Some were overcome by making some assumptions and others were accepted. These are explained below.

1. The data is not as rapid as it would come as in a fire hose of twitter, this restricts a user’s accessibility to the complete information that he desires.

2. Dealing with personalized trends means that the user set being dealt is small and restricted. We might not have all classes of users in the dataset. We are actually doing sub-classification rather than working with a generic set of users. Hence, it needs to be assumed that the trending topics might actually be applicable for a certain group of users.
This is expected and desired. Also, it is worth mentioning to assume that users dealt might fall into only one category as a whole in some cases.

3. Life time of each trending topic is more in Personalized trending topics because of the relatively low flow of tweets and people in a small group tends to update lesser tweets as compared to the global inflow.

4. There might be some stagnant trending topics in a group which can stay for a long time and never ever go out because of the group mentality might be the same.

Example: Even when seen in twitter, twitter has to remove Lady Gaga and Justin Bieber [4] because they were always staying on the top of the trending topic list because of its rate of occurrence and constant ReTweet!

5. We use Twitter’s data in the following problem by using the Twitter’s API. All the data we calculate here are assumed to be obtained in real-time and hence we in turn depend on twitter’s API to provide us with the latest information.

2. Background

2.1. Twitter

2.1.1. Overview of Twitter

Twitter [5] currently holds the largest traffic of information in terms of the number of updates compared to other competitive social networking sites such as Facebook and Google plus with about 1 to 4 updates per hour per user [6]. The growth of micro-blogging sites such as Twitter has further influenced the rate at which the sites are being used. Even with a meager 140
character limit blogging service, Twitter, now has the lion’s share of information flow in the social networks. The 140 character limit enforced by twitter makes user to limit their message to summary of the news. Though this is good, for most of the cases a tweet will include a link or two to the actual news of the message. Appropriate to the small message limit in twitter, each message in twitter is called a “tweet”. We will also use tweet as a common term throughout the thesis to represent the messages sent through twitter.

The message limit in Twitter is enforced due to the limit in the Short Text Messaging (SMS) which has the same 140 character limit. This also makes it easy for users to update twitter updates through their mobile by just sending out the message as a text. This adds to more overhead of already full database servers.

With a daily flow of about 200M tweets or about 2300 tweets per second (Data as of July 2011) it is a cornucopia of useful data. This inflow of data also provides a better foundation and a larger set of data to analyze and digest. Mining Twitter has become a vast research area and identifying interesting data is one of the major sought after problems. We also discuss about various approaches that people have used to analyze Twitter. We have discussed many terms in the introduction about which we would be talking about in the next sections.
2.2. Concepts and terms

2.2.1. Trending Topic

A trending topic has various definitions depending on the context and the usage of the term. We define it in a couple of ways which complement each other to tell about the definition of the trending topic.

A “trending topic” is a word that can represent the context of the tweet. It can be the topic of the discussion or it can be the important word that weighs the message.

A “trending topic” is a word that occurs repeatedly over a period of time in various tweets defining its importance. Since a trending topic can represent a current event or a topic that people
are interested in talking about. This also makes the trending topic very volatile and can vary very quickly from one to another.

Trending topics climb to the list either by a mass revolution or interest like “Iranian Election”, “csisimportant” etc. or because the topic of discussion is interesting like “iPhone5” or “funnyjokes” etc.

2.2.2. Circles

Any social network is created to form a platform to create a communication between different individuals. As the duration of communication increases, there would be a polarized shift of individuals being attracted slowly to their like-minded counterparts. As like-minded people get together, they tend to form groups in any situation.

These groups can form in various ways. In Twitter, a user follows another user if the user likes the tweets of another user. This process is called “Following” of user X on user Y. Similarly, user X is called the follower of user Y. The group here is the set of users who the user follows and the user, since the user believes that his tastes meet with some users and follows them.

We use “Circles”, “Groups” and “Neighborhood” interchangeably because different social networks use different terminology to represent the same concept. Ultimately it all means that a group contains a set of users who are treated as best matches for a single user X to discuss various subjects of him. So, this is called as the circle of the user X or the neighborhood of user X.
2.3. Related Work

Most of the work related to twitter and trending topics are related to life span of trends and trend analysis. As defined above, trending topic is very volatile and has comparatively short life span when compared with blogs or updates on Facebook.

Birth of Twitter topics: Twitter topics are nothing but updates that may be posted based on true facts or can be completely fabricated news. Twitter carries a lot of information around through internet. Though most of the information is lost in the flow, yet it contains crucial information regarding the most happening updates. For example, twitter played a major role in Egyptian revolution [7] in bringing down Mubarak’s regime, where people were completely isolated from most of their communication mechanisms and were lucky enough to communicate through Twitter via Internet. If it was not for Twitter, the revolution in Egypt might have been suppressed without awareness of their support in other countries.

As Twitter’s trending topic detection algorithm stays in the dark there are a few works in literature which try to identify trending topics in twitter. Identifying trending topic using the actual data of twitter is quite impossible with given resources for the reason that, twitter does not allow any publishing of results derived from the twitter data. Secondly, the actual data used by twitter or “FireHose” [8] is only sold for business purposes. But, this can be fairly approximated by using the small stream of data or “GardenHose” that can be obtained through the Twitter’s APIs. As explained above, we use User Stream APIs and REST APIs of twitter to reach the similar goal. As most of the literature restricts in using the APIs there are some methods where
3. Implementation

This section will explain about the implementation part of the thesis. It briefly explains the tools and language used in coding. Starting with the overview of how the system works, we walkthrough the code snippets and other parts of open source code used for the implementation.

3.1. Technology Used

3.1.1. Coding: Java

Java is one of the most used programming languages especially for a cross-platform machine independent type of software. It offers wide variety of features like Object oriented programming, Powerful class library and easy connection to various databases. The main reason behind choosing java is because of various available libraries that can access Twitter, parse JSON (Java Script Object Notation) format which is returned by Twitter.

Java is easily extendable to various other platforms and also, java software can be easily extended for any further developments.

3.1.2. Access to Twitter API: Twitter4j

Twitter4j is a Java library for Twitter API. It can be easily integrated into any kind of java application from android apps to desktop applications. It provides user with a wide variety of code example and also has a detailed documentation.

3.1.3. Backend: MySQL

MySQL is one of the most widely used relational database systems especially for web applications. It is a scalable system that is being used by large companies like Google and
Facebook. MySQL provides easy setup and simple accessibility that made it the top choice for this implementation.

3.1.4. IDE: Net beans

IDE to develop the code, establish internet and database communications. Also, Net Beans served as a Source Code repository to backup code of this implementation.

3.2. Flow Charts

3.2.1. Pre-requisites

The important step for proper functioning of the software is to set the user credentials in the Twitter4j.properties file or by setting them explicitly as mentioned in the Implementation part. Once the credentials are validated, further process is carried out. The credentials are not a mandatory part, if the credentials do not validate, the application rolls back into the admins account and tries to retrieve only public information. The information that is set to private and that cannot be accessed by the admin account would be inaccessible and would remain out of scope to analyze.

3.2.2. Stage I – Initial Data Collection

This is the most important phase of the software where it collects required data from Twitter.
Before the software starts to give out trending topics for the user, it pre-populates all the information that can be obtained regarding the user into the database. It parses the received tweets and sets up the platform. All the data at this stage is obtained through the REST API of Twitter which allows a maximum of 350 requests per hour.

### 3.2.3. Stage IIa - Word Extraction

All the tweets belonging to this user are retrieved from the database. Each tweet is then parsed into words and individual database is automatically created for the user and the words that are broken up are stored in it.
3.2.4. **Stage IIb - Feature Extraction**

Stage IIb works hand in hand with Stage IIa to provide on the fly feature collection for each word. The first step is to eliminate the stop-words that are presumed not to form any kind of trending topics. Next step of Feature extraction is to analyze each individual word and extract the tweet level features and word level features. These are explained in detail in further chapters. All the scores from every feature are stored in database.
3.2.5. Stage IIc – Trending Topics

This step has all the business logic that calculates and ranks the trending topic based on the features extracted. More details on how the trending topics are calculated are given in further chapters.

![Trending Topics Diagram](image)

3.2.6. Stage III – Real-time data collection

This step is much similar to Stage I, with the only difference of collecting the tweets from a streaming API rather than the REST API. Stage III mainly involves the streaming API listener and the caller to the Stage II process. Streaming API provides us with the up-to-date real-time data of a user and tweets that are updated by his friends. Once a new tweet is posted from any of the neighborhood of the user, Stage II is called and the process is automatically completed.
3.3. Significant Features: Tweet Level Features

3.3.1. Time Stamp and Epoch Time

In tweets, time stamp is of the most important feature of interest. A tweet that is recent holds a lot of importance and weight compared to a tweet that every tweet that is being carries a timestamp of its generation. These timestamps play a crucial role in deciding whether a word is becoming a Trending Topic. A topic becomes trending when the word starts occurring repetitively in latest tweets. The more the topic repeats the better is the trending topic gaining ground.

But, the scenarios of trending topics are not that straight forward in some situations. To effectively identify a Trending Topic we need to utilize our earlier knowledge of Trending Topics. To better explain the scenario let us take an example.
In the above situation, it shows that in Burst 1, Word 2 has occurred for some good time and must have been a trending topic (compared to word 1). After some time, Word 2 starts occurring intermittently but often enough that it can be a trending topic again. Word 1 on other hand starts to occur in tweets and is trying to get onto the trending topic list. Who should win in such situation? Intuitively, we can say that Word 2 should win over Word 1 because it has already occurred earlier and it has higher probability to become a trending topic again in future.

And consider a second situation like this,
Consider the above situation which is similar to the earlier one. Though the order of bursts is the same, the only main difference is the Burst 3 which occurs more densely than the earlier situation. Though Word 2 has earlier been a trending topic, the more dominant of the both words in the recent times is Word 1. Hence, Word 1 should prevail over Word 2 in terms of being a trending topic.

In this thesis, we solve the problems explained above using the following technique. The best solution we found for the above problems by utilizing the advantage of prior knowledge is by using an Epoch Time.
Epoch Time is a timestamp of an event or an occurrence. But, in this case we use the Epoch time as a time stamp which is the starting point of the current trending topics age. Say, a topic X is trending currently, it means that the topic X is a trending topic from the Epoch time.

Now, this epoch time is used as a reference for the time at which the tweet occurs. All the words in the tweet are considered to be occurred at a time equal to

\[ T = t_1 - E \]

Where, T is the calculated time and \( t_1 \) is the actual occurrence and E is the Epoch time.

This equation gives a two-fold benefit,

Firstly, it reduces the timestamp to a difference of timestamps which is a number and can be easily modified to store and compare with the weights obtained by the other features.

Secondly, it can be used to eliminate stop words and also provide and determine the trending topics in a sophisticated fashion.

The placement of Epoch time does not matter as we are just calculating a time difference. (A reference to the above figure and a tweak made is explained in “Stop Word Removal” section. Also, the reason how stop words can be eliminated is explained in that section).

When the same word occurs again at another instant, the above process is repeated and the time difference is added.

### 3.3.2. Mentions Score

Twitter provides a feature to include any user’s name in a tweet by just adding a ‘@’ before the username. Doing it would send a notification to the respective user and displays the tweet in
corresponding user’s home page. This feature can be exploited to suggest us a better trending topic. There are two cases when a user might be mentioned by another user.

1. When a friend believes that user might be interested in the topic of the tweet. Example: “@Bob: Check out the updated Netflix app for your Captivate - http://engt.co/mQoiDO”

2. When a friend want to communicate with the user privately. Example: “@Bob: How about coffee at Starbucks in an hour?”

In either case, it is an additional suggestion that tweet is highly related to the user. Hence, all the words occurring in the tweet with current users mention are given a score.

### 3.3.3. ReTweet Score

When any user likes a particular tweet, he can share the tweet in his own profile, this is called ReTweeting. Twitter provides us the information about how many times a tweet is ReTweeted. The higher this number is the higher is the popularity in the particular tweet. Even if in user neighborhood, this is not a popular tweet, it might gain importance quickly because of its wide acceptance. Factors that increase this score:

1. If there are more people who ReTweeted the tweet, then the score would be higher

2. If the tweet is a heavily ReTweeted tweet in the rest of the web, and it is ReTweeted by at least one person in the current user's friends.
3.3.4. Favorite Score

If a user marks any tweet as his favorite it would mean that he has a special interest in the particular tweet. A user generally favorites when he thinks he might want to revisit that context again. Hence, favorites are another feature that can enhance the rank of a topic. The rate of increase of this score might be small, but it can suggest us really interesting topics of the user. Generally, the range of number of favorites varied from 0 to 27 in our dataset.

3.3.5. Hop Distance

Hop Distance is the distance of the tweeted user or friend from the user. A user itself will have a distance of 1 and his friends would have a distance of 2. This is used as a factor of scalable component. When user can have the capability to select the neighborhood diameter, he can choose the distance to which he can cover the tweets.

3.3.6. User Authority Score

This is an inspiration from Kleinberg’s HITS Algorithm [9] to find the user authority score based on the number of followers a user has to his profile. This would suggest us the importance of the user, and his tweets can be assumed to have a similar score. This is because users with more number of followers will have more important tweets and would be ReTweeted by more users and are more authoritative.

3.4. Significant Features: Word Level Features

After identifying the tweets that have higher feature weight based on the above statistics may still be overwhelming. We actually need to get words that are of importance rather than the whole tweets. Each word is attached with its appropriate word level feature and is given weight
based on its feature. To begin the classification, we started with some pre-determined features that are of interest.

- Word Count
- Damping Factor
- Hash Tags
- Global Trends
- URL

These terms will be defined in detail in the following section. Each word in the tweet will then be classified based on these features; terms that come out of this pass with a better weightage would be treated as a trending topic.

3.4.1. Word Count Score

Word count is the raw score of the number of times the word occurs in the set of tweets in database. Though this is one of the most important features to keep a tag on, yet this can be a very poor indicator of how trending a word can be in the tweets. Hence, it is equally important to apply other feature areas to decide on the trending topic.

3.4.2. Word Types: Hash Tag / General Words

Twitter defines hash tag as the keywords that are appended by symbol ‘#’ which have high tendency to become a trending topic [10]. Though hash tags have better probability to become trending topics, other words that are not tagged with the symbol ‘#’ are also equally important to consider as trending topics. In our scenario of personalized trending topics we consider the words with ‘#’ attached to it to have more probability of becoming a trending topic.
General words are those without the ‘#’ included as a prefix to them. Other than the additional weightage the hash tag gives to a word, all other words are treated equally important as hash tags.

### 3.4.3. URL

Many tweets are actually a summary of the actual news that is posted somewhere else. And it compels users to post URLs containing the actual news or link that they are talking about in the tweet. URLs are mostly shortened using the URL shorteners as putting to the actual URLs in the 140 character limit is not feasible. URL shortening is a service provided by many third party websites such as bit.ly [11] and tinyURL [12] including twitter [13].

There is an importance associated with URLs when calculating trending topics. A tweet containing an URL is most likely to be one of the trending topic related tweet [14] [15]. Hence URL is also considered as a separate feature weight of the word. This also means that an URL can itself become a trending topic competing with the other words.

There is a problem detecting similar URLs especially when they are masked with the URL shorteners. Luckily there is an easy work around by utilizing the meta data of the URL that twitter provides. Twitter contains the actual URL, or the expanded form of the URL with the tweet information that we receive through API. Hence we will be able to actually assign importance to the URL itself.

### 3.4.4. Damping Factor

By definition of trending topic, it will never stay at the top for too long. And if it does stay for too long as a trending topic it might as well mean lead in decline of interest in that topic. Twitter
has been trying to censor some of the trending topics that are on the top list for more than a certain period [4]. During early 2010, with “Justin Bieber” and “Lady Gaga” topping the trending topics for too long, twitter censored the topics to remove them from the trending topics chart to create more space for the trending topics that are more current and interesting. The same process needs to be applied to the personalized trending topics.

The damping factor does this work for us by reducing the effect of too stale trending topics in the top charts. This is also a convenience factor that can be used on certain stop words which are not filtered by the noise filtering techniques. For local trending topics, there might be some terms which are specific to their neighborhood like College specific terms or local slangs. These terms needs to be eliminated by applying the damping factor explicitly. Damping factor is not calculated through algorithm but applied based on the time a topic stays on the top list. This is a convenience feature introduced later in the stage to provide more up-to-date results of trending topics.

3.4.5. Global Trending word

Twitter already publishes the trending topics on a global scale. These are the latest topics or words that are trending on a global level. We can take the advantage out of the global trending topics by knowing what kind of words have a tendency to become trending topics quicker than the more common words. Using this information we assign more weight to the words which have occurred in the global trending topics list. Each word that occurs is checked with a cached global trending list. If the word matches one of the global trending lists, then it would be added the flag that it belongs to a global trending list.
3.4.6. Calculating the trending topic

After calculating various feature weights for the words, we get a list of different weights on various levels. These weights can be normalized to form a set of even weights which can then be applied to the following function. The problem that remains in calculating the trending topics is more of a ranking problem which can be defined as a function of three variables,

\[ \text{Trending topic} = F(t, w, f) \]

Where,

- \( t \) is the time stamp or the epoch difference of the word,
- \( w \) is the weight accrued in the feature level filters,
- \( f \) is the friend who tweets in the following group of the user

4. Experimental Setup and Results

4.1. Data

The most important part of the work is collection of data and maintaining of data. While the exact details of the store and database are excluded from this section, there are additional details about the database and architecture in the Appendix. The dataset used in our experiment is collected through the Twitter’s public APIs [16]. These results were built with data collected between February 2011 and July 2011.
We collected data for random sets of users whom we can validate with the followers and tweets about the trending topic in their neighborhood. Each set of tweets obtained through the Twitter API contains the various details that are provided in the Appendix.

Unfortunately, Twitter does not provide us with a full flow of Twitter feeds because of privacy issues.

4.2. Code

We used java to develop this solution and all the details of the coding and the database part is moved to the Appendix section of the thesis.

4.3. Word Analysis

4.3.1. Distinct words

The first step of the process is to read the tweets of a user and split them up into words that can be further processed for the features. We take an example walkthrough of a case in our thesis.

In this current dataset we have about 65K tweets in the neighborhood of the user.

When the Word splits are done into individual words they split up to 104K distinct words. These words would contain all kinds of words ranging from stop words to the words which can potentially be the trending topics. These words should eventually get down in number and we should be able to find out the actual ranking of each of the words.

<table>
<thead>
<tr>
<th>Type</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Tweets analyzed</td>
<td>64229</td>
</tr>
</tbody>
</table>
4.3.2. Top used words

Before we actually apply filters on the data obtained, we tried to analyze the kind of words that are potential and how they are distributed. When the words are arranged according to the number of times it was repeated.

![Average of Top-35 used words (including stop-words)](image)

*Figure 11: Graph of Top-35 words*

From the picture we can interpret that the words occur the highest are the common words like “a”, “an” and “the”. These words that convey no meaning in the trending topics are called the stop words.

There are two types where we can remove the stop words.
1. Have a dictionary of high frequency words that include the most commonly occurring words in twitter. Each word can be checked against this dictionary and can be removed if the word is one of the stop words.

2. The second method is to use an epoch

The advantage of using an epoch is that it can also eliminate the stop words that occur continuously over a time period this technique is explained in the “Stop word removal” section of the thesis.

<table>
<thead>
<tr>
<th></th>
<th>Word</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The</td>
<td>15.48498</td>
</tr>
<tr>
<td>2</td>
<td>You</td>
<td>6.875518</td>
</tr>
<tr>
<td>3</td>
<td>And</td>
<td>6.251117</td>
</tr>
<tr>
<td>4</td>
<td>For</td>
<td>6.046266</td>
</tr>
<tr>
<td>5</td>
<td>With</td>
<td>2.887983</td>
</tr>
<tr>
<td>6</td>
<td>That</td>
<td>2.743682</td>
</tr>
<tr>
<td>7</td>
<td>This</td>
<td>2.336524</td>
</tr>
<tr>
<td>8</td>
<td>Your</td>
<td>2.270727</td>
</tr>
<tr>
<td>9</td>
<td>Are</td>
<td>2.162215</td>
</tr>
<tr>
<td>10</td>
<td>Have</td>
<td>2.079833</td>
</tr>
<tr>
<td>11</td>
<td>All</td>
<td>1.918722</td>
</tr>
<tr>
<td>12</td>
<td>Can</td>
<td>1.804858</td>
</tr>
<tr>
<td>13</td>
<td>What</td>
<td>1.791494</td>
</tr>
<tr>
<td>14</td>
<td>From</td>
<td>1.695859</td>
</tr>
<tr>
<td>15</td>
<td>Out</td>
<td>1.686738</td>
</tr>
<tr>
<td>16</td>
<td>Just</td>
<td>1.671245</td>
</tr>
<tr>
<td>17</td>
<td>Not</td>
<td>1.644712</td>
</tr>
<tr>
<td>18</td>
<td>Love</td>
<td>1.615281</td>
</tr>
<tr>
<td>19</td>
<td>New</td>
<td>1.608151</td>
</tr>
<tr>
<td>20</td>
<td>But</td>
<td>1.475764</td>
</tr>
</tbody>
</table>
The average of the words is calculated based on the following equation:

\[
Average = \sum_{i=1}^{n} \frac{N_i}{F_i}
\]

N_i = Number of word occurrences for User i;

F_i = Number of Followers for User i;

n = Total number of users;

4.4. Final Results

The words obtained from the previous stage are then stored into a database. These words are first processed through the stop word removal mechanisms, where most of the stop words are discarded into the garbage collection. After obtaining the filtered words from the earlier stage, we apply the rest of the filters. The rest of the filters involve in assigning weights for each of the words based on various features discussed above.

In the next sections, we start discussing the importance of various tables generated as bi-products during the result.

4.4.1. All-time top contacted users in the Circle

Once the weights are obtained for various features for each of the words, we can derive various results from it, one such kind of result is isolating the users in the words. In the following table we can see various users of the current user ordered according to the level of rank they obtained. “@engadget” in this example has the highest contacted user rating and where as
“@funnyoneliners” has the fourteenth position. These are the users that the current user is trying to tweet about. Or during tweets, the current user is mentioning mostly about these users based on their ranks. This is a useful metric to know the level of interest in each of the user in his friends.

<table>
<thead>
<tr>
<th>word</th>
<th>wcount</th>
<th>wordType</th>
<th>fromMention</th>
<th>fromFavorite</th>
<th>fromTweet</th>
<th>hcpDistance</th>
<th>timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>@engadget</td>
<td>19</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>172821544649</td>
</tr>
<tr>
<td>@MadhuSudhananJ</td>
<td>14</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-236036030302</td>
</tr>
<tr>
<td>@provostthmo</td>
<td>14</td>
<td>2</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-12911981842</td>
</tr>
<tr>
<td>@dkarthik</td>
<td>12</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>22793553184</td>
</tr>
<tr>
<td>@sellisto</td>
<td>10</td>
<td>2</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-82660979872</td>
</tr>
<tr>
<td>@KatherKvelyn</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-310968927467</td>
</tr>
<tr>
<td>@brianthecoder</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>15694393700</td>
</tr>
<tr>
<td>@saronrankin</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-9927019203</td>
</tr>
<tr>
<td>@OREillyMedia</td>
<td>8</td>
<td>2</td>
<td>82</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>16833510855</td>
</tr>
<tr>
<td>@the4subpro</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>649220461</td>
</tr>
<tr>
<td>@saadotopen</td>
<td>8</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>7392360358</td>
</tr>
<tr>
<td>@proffreda</td>
<td>7</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-7552671014</td>
</tr>
<tr>
<td>@longint</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>465965201</td>
</tr>
<tr>
<td>@funnyoneliners</td>
<td>6</td>
<td>2</td>
<td>606</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>8796648265</td>
</tr>
</tbody>
</table>

4.4.2. **Trending users in the Circle**

This is the list of users obtained when we query for trending users. The difference between most contacted users and trending users is that, most contacted users are those who are being contacted by the current user repeatedly.

Trending users are the users who post topics that have tendency to go into current trending topicing list. These are the list of the users in the current user’s neighborhood and post the most trending topics.
4.5. Trending Topics

These are the final results we obtain for various users we have tracked through our implementation. Each user’s name is given on the tabs below, and their respective trending topics are shown below in their tables.
If we observe closely we can find some of the features taking the role clearly in the tables.

For example, Murdoch’s case was one of the most prominent global trending topic at that time and we can see that Murdoch was repeated consistently over various users. Also, every users fields were properly identified by the corresponding terms in each of their fields.

### 4.6. All-Time Trending Topics

All-time trending topics are similar to the trending topics but with a lower Epoch value. As explained, a lower epoch value would say that the starting age from which the topic is big. This means that the trending topics are being calculated for a larger period of time. Thus Epoch time proves useful again in finding the All-time trending topics of users.

![Table of trending topics](image)

Most of these results are too personalized to realize that the results are correct. Hence the users displayed here are from some known personalities so that it is easy for comparison.
5. Future Scope

5.1. Using TF-IDF

Term frequency – Inverse Document Frequency is one of the techniques to identify the weights of the terms as they occur in a document or for our case a neighborhood. In the thesis, though we used some techniques to replace TF-IDF, it would be a good enhancement if the actual utilized some of the concepts of TF-IDF to eliminate stop words [17]; it has not used the complete advantage of TF-IDF. It would be a good way to work towards full-fledged implementation of the TF-IDF technique to eliminate the stop words.

5.2. Increasing the neighborhood

This thesis stops at a point where trending topics are identified for a neighborhood with a 1-hop distance. That is, a user gets to know the trending topics in his “following” list of twitter which is referred as 1-hop. This can be extended to a multi-hop approach by just increasing the number of hops in the implementation. By 2-hop we mean that the data is collected from the users who are followed by the user and the users who are followed by user’s “Following” users and vice versa.

It can also be thought as, if we sufficiently increase the number of hops or do a multi-hop repeatedly and eliminate the loops we would eventually get to a point where the data contains all the tweets globally and applying the technique would yield the global trending topics instead of personalized ones. As we are not interested, currently, in either finding the global trending topics or a multi-hop tweets which would dilute the actual trending topics. We restrict ourselves to the given features and single hop and to the user in the twitter. But certainly it is a great way to proceed in future.
5.3. Using machine learning techniques

Support Vector Machines or SVM are the concepts that are used in recognizing patterns of data using supervised machine learning methods. In the thesis, we have identified the trending topics of a group which can be of interest to a user. Using this data, we can train an SVM to teach it to identify similar patterns of data in the tweets. Once the SVM is developed to a certain extent, it can then be applied on the actual twitter stream to identify tweets that are interesting to the group. This is a great way to increase the approach of a user in the social network. A user not only gets the trending topics inside his group, they can also gain the access of interesting tweets from the other parts of Twitter. A RankSVM can be used [17] [18] to evaluate the ranks of the trending topics. Yajuan et al. tried to rank tweets using the SVMs and have reported decent success in their approach in this regard. Though their technique ends in identifying and ranking interesting tweets, it can be extended to detect and identify interesting topics inside those tweets. Analysis of what algorithms to apply and how to optimize the SVM will be left as a future scope of study.

5.4. Aggregating and forming new trending topic groups based

As we have developed trending topics individually for each user, we can extend it to cross user applications by aggregating the topics. The topics we have for each user reflects their personality. If we need to find common details across a group of individuals, we can also group the personalized trending topics and find the intersection between them to get the overall picture of the group of users. We can have common topics between users and we can aggregate users based on the trending topics. We believe that this is a different approach in knowing personal choices of a set of users.
6. Conclusion

The work described in this thesis works as a foundation towards further research. Finding personal user trending topics is a relatively new area of study. User trending topics are very useful and helpful in various perspectives as discussed in the initial stages of this thesis. This is primarily because we are able to understand a person at a more granular level.

Existing work is mostly done in the areas of Trend Analysis, which describe how the trends are created and the reasons behind the growth of trends. Though this provides an insight in understanding the growth of trends it does not provide us with actual information that can be extracted from the online social networks. This work is an attempt towards filling this gap of trend analysis and the trending topic creation. We discussed in detail, some ways of analyzing the features that influence the tweets and the stop words that need to be removed from the tweets to get trending topics.

- In the introduction, we have discussed the motivation and the importance of the problem statement we stated.
- Next, we provided fundamentals that can help the reader to understand the terms and concepts used in the thesis.
- Thirdly, we provided two sections of Implementation that are involved in identifying the personalized user trends.
7. Bibliography


[22] Jeff Story, Jason Wickstra, “Discovering Trending Topics on Twitter Via Retweets.”. ACM, April 2009


[31] Analysis and classification of Twitter Messages, Christopher Horn, April 2010, Master's Thesis at Graz University of Technology.

[32] Time-aware and trend-based authority ranking, Klaus Lorenz Berberich.


[34] Beyond Trending Topics: Real-World Event Identification on Twitter, Hila Becker, Mor Naaman, Luis Gravano


[36] Want to be Retweeted? Large Scale Analytics on Factors Impacting Retweet in Twitter Network, Bongwon suh, Lichan Hong, Peter Pirolli, and Ed H. Chi
8. Appendix A

8.1. Pre-requisites

A user name and password has to be provided to the application. Since, we are using a java library for authentication; the details are directly submitted to the library. These details can be provided in different ways.

8.1.1. Storing a configuration file: twitter4j.properties

```plaintext
debug=true
oauth.consumerKey=***************************
oauth.consumerSecret=****************************
oauth.accessToken=****************************
oauth.accessTokenSecret=*********************
```

Code Snippet 1: Configuring through .properties file

8.1.2. Configuring through the Command line arguments

It can also be done by providing the details through the program from the command line.

```
> java StartHere <user_name> <password>
```

Code Snippet 2: Configuring through command line arguments

8.1.3. Authenticating

Using the following code we gain access to the twitter’s REST API by authenticating the user credentials. In the following code ‘cb’ is the configuration obtained from the above steps.

```java
Twitter = new TwitterFactory(cb.build()).getInstance();
```

Code Snippet 3: Authentication
8.2. Stage I – Initial Data Collection

8.2.1. Get users timeline

The following code provides us with the statuses of “currUser” retrieved from the REST API of the Twitter. Where, ‘twitter’ is the REST API object obtained after authentication.

```java
twitter.getUserTimeline(currUser);
```

Code Snippet 4: Get Users Timeline

8.2.2. Get follower Ids

Now in the next step, we retrieve the user IDs of each of the person the current user is following. We will refer to the ‘following’ users as friends.

```java
IDs followers = twitter.getFriendsIDs(currUser.getId(), cursor);
```

Code Snippet 5: Get follower IDs

8.2.3. Get followers timelines

For each of the ID we obtain in the above step we would perform a similar operation as retrieving the users timeline, to retrieve the statuses of each of the friends.

8.3. Stage IIa - Word Extraction

8.3.1. Lexical Analysis

There are two steps of lexical analysis that is done while extracting words. First level of word extraction involves in identifying special attributes of tweets like HashTags, Mentions and URLs that may be present in the tweet. Second level involves in removing special characters (including
smilies and punctuations). To do the lexical analysis, java has one of the best tools called “String Tokenizer.” Some interesting code snippets are given below.

URL detection is done by using java’s inbuilt URL class. When we try to attach a string to URL class, if it’s a proper URL then it’ll create a URL object successfully otherwise it will return a MalformedURL Exception.

```
Try {
    URL url = new URL(level1Token);
    //Rest of the code for URL
} catch (MalformedURLException ex) {
    //Code if the level1 parsed word is not a URL
}
```

**Code Snippet 6: Identifying URL**

The second level, as explained before involves stripping off all the special characters from the tweet and extracting individual words. The code for removing all the special characters,

```
StringTokenizer level2 =
    new StringTokenizer(level1Token, "\"\",=;%&*(){}+=!:\":;\n;?)
    // Not Parsing Symbols @ # \ / ?
```

**Code Snippet 7: Stripping off special characters**

### 8.4. Stage IIb - Feature Extraction

Feature extraction is the critical point, where each of the feature is categorized into two levels and are analyzed and the corresponding ranks are stored in the trending topics table. The theory behind feature extraction is explained in a separate section.
8.4.1. Tweet level features

These are the features of a tweet itself and remain constant for all the words of the tweet. Example: Timestamp. So, these are recorded before the tweet is sent for word extraction.

8.4.2. Word level features

A word is again classified into different categories based on the type of its occurrence and location. Example: word count.

8.5. Stage IIc – Trending Topics

This stage involves in applying of the business logic that will be explained in the feature extraction chapter followed.

8.6. Stage III – Real-time data collection

Once all the data is initialized and initial run was successful, this stage sets in. It involves a stream listener that listens to the stream of the user and raises an event whenever a new tweet is posted. The sample code for this is given below,

```java
static UserStreamListener listener = new UserStreamListener()
{
    public void onStatus(Status status) {
        // Call Stage II
    }
}

TwitterStream twitterStream =
new TwitterStreamFactory(cb.build()).getInstance();
twitterStream.addListener(listener);
```

*Code Snippet 8: Stream Listener and Stage II entry*
At this stage, it is worth mentioning that all throughout the code no dependent functions were written. This allows us to execute Stage II process as a multi-thread function. This will enable us to deal with the high stream flow efficiently using threads.

### 8.7. Exception Handling

Twitter has a rate limiting over its REST API and allows only 350 user requests/hour and it completely discourages repetitive requests when the rate limit has reached. There is a chance of blacklisting the user from any future API requests. Hence, appropriate exception handling was placed to allow the software to wait for 20 minute intervals if a request was denied because of rate limiting.

```java
try {
    System.out.println("Problem communicating with Twitter:");
    if (te.getMessage() == null) {
        System.out.println("\tNULL Exception");
    } else if (te.getMessage().startsWith("401")) {
        System.out.println("\t401");
    } else if (te.getMessage().startsWith("400")) {
        try {
            System.out.println("\tRate Limit Reached! Waiting…");
            Thread.sleep(1200000);
        } catch (Exception e) {
            System.out.println("Cannot Wait: " + e.getMessage());
        }
    }
} catch (TwitterException te) {
    System.out.println("Problem communicating with Twitter:");
    if (te.getMessage() == null) {
        System.out.println("\tNULL Exception");
    } else if (te.getMessage().startsWith("401")) {
        System.out.println("\t401");
    } else if (te.getMessage().startsWith("400")) {
        try {
            System.out.println("\tRate Limit Reached! Waiting…");
            Thread.sleep(1200000);
        } catch (Exception e) {
            System.out.println("Cannot Wait: " + e.getMessage());
        }
    }
}
```

**Code Snippet 9: Exception Handling - Rate Limit Reached**

### 8.8. Design Patterns: Singleton Pattern

Throughout the code, creating a database object or a twitter object uses a special function called getInstance(). This is taken from one of the design patterns in programming called Singleton Pattern. A class diagram of current “Database Access Object.”
8.9. Database Organization

8.9.1. Tables for Stage I and Stage III

8.9.1.1. Tweets

Stores every tweet received into the following table. It has the tweet id as primary key.
8.9.1.2. Users

Each tweet that we receive, contains the details of the user that

8.9.1.3. Followers

This table contains the relation between each user and their corresponding friends (the people who follow the current user). It also contains the user authority of the friend.
User Authority: There might be a case where a user might have a common friend with another user whom we are tracking. In this case, we have two same follower_id with different user_ids. User Authority changes for a person based on which neighborhood we are calculating. Hence, it was decided that this is the right place to store the user authority.

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>user_id</td>
<td>BIGINT</td>
</tr>
<tr>
<td>2</td>
<td>follower_id</td>
<td>BIGINT</td>
</tr>
<tr>
<td>3</td>
<td>userAuthority</td>
<td>FLOAT</td>
</tr>
</tbody>
</table>

Figure 15: Followers Database Table

8.9.1.4. Hashtags

Each tweet can contain multiple Hashtags, this table is maintained to understand the hashtag flow in tweets. It is also used to track back the origin of the hashtag to the corresponding tweets.

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>tweetId</td>
<td>BIGINT</td>
</tr>
<tr>
<td>2</td>
<td>Hashtag</td>
<td>VARCHAR</td>
</tr>
</tbody>
</table>

Figure 16: HashTag Database Table

8.9.1.5. URLs

Similar to Hashtag table, URLs from each tweet are identified and extracted and are stored in the URL table. These URLs can be tracked back to find the origin of the tweets. Also, this table is very useful when we extend this program to gather more information regarding the tweet based on URLs.
8.9.2. Tables for Stage II

8.9.2.1. Trending topics table

After parsing each of the tweets, when each feature is extracted it is promptly inserted into this table which holds different scores based on different features as explained in the feature extraction section. The final score is stored in the ‘trendRank’ column of the table. This column is updated everytime a new trend is received and hence it stays up-to-date with the current trends stored in the database. Showing the Trending topics is straight forward from this table. Querying for the Top ‘trendRank’ would give us the top trending topics of the tweets.
8.9.2.2. Stop words table

This is the place where we place the words that are to be eliminated from the trending topics. Initially, it contains the stop-words that are the most frequently occurring words in tweets and which are not of interest to our current research. If certain trending topics are to be eliminated, adding a new tuple to this table would do the task of eliminating the particular trending topic.

8.10. Structure of a tweet

```
{
    "entities":
    {
        "hashtags": [],
        "user_mentions":
        [
            {
                "indices": [3,17],
                "screen_name": "YourScreenName",
                "id_str": "40884854",
                "name": "Name1 Name2",
                "id": 40884854
            },
            {
                "indices": [62,71],
                "screen_name": "filmfare",
                "id_str": "35695228",
                "name": "Filmfare",
                "id": 35695228
            }
        ],
        "urls": []
    },
    "text": "RT @YourScreenName: For those who care, as far as magazines go @filmfare has aced it on twitter. We marched a few thousands ahead of hol ...",
    "id_str": "81242112211914888",
```
For those who care, as far as magazines go @filmfare has aced it on twitter. We marched a few thousands ahead of hollywood reporter.
"following":null,

"profile_background_image_url":"http://a.twimg.com/images/themes/theme1/bg.png",

"description":"Editor, filmfare. loves movies, music, books. will not tolerate rudeness. kindness moves me. Lead the life I love. Keep the faith. Spread happiness",

"favourites_count":1,
"created_at":"Mon May 18 14:36:17 +0000 2009",
"profile_text_color":"333333",
"is_translator":false,
"show_all_inline_media":false,
"geo_enabled":false,
"profile_sidebar_fill_color":"DDEEF6",
"default_profile":true,
"listed_count":490,
"profile_background_tile":false,
"friends_count":88,
"protected":false,
"contributors_enabled":false,
"followers_count":24802,
"name":"Name1 Name2",
"statuses_count":18400,
"profile_link_color":"0084B4",
"id":40884854,
"verified":false,
"utc_offset":-36000,
"profile_sidebar_border_color":"C0DEED",
"url":null,

"profile_image_url":"http://a.twimg.com/profile_images/134003186/jiptpil Normal.jpg",

"id":81234001356001280,
"geo":null,
"user":
{

   "default_profile_image":false,
   "lang":"en",
   "notifications":null,
   "time_zone":"Mumbai",
   "id_str":"58897898",
   "profile_use_background_image":true,
   "screen_name":"rahulnanda86",
   "profile_background_color":"CODEED",
   "location":"
   "follow_request_sent":null,
   "following":null,

   "profile_background_image_url":"http://a.twimg.com/images/themes/theme1/bg.png",
   "description":"Senior Multimedia Executive. Politically straight, music buff, cynical, dyslexic, wannabe photographer and filmmaker."}