Twitter Trends

A Project Submitted
in Fulfillment of the Requirements
for the Degree of
Bachelor of Technology
in
Information Technology

by
Group 1 (B.Tech 8th Semester, IT)

Medha Srivastava (20128048)
Rohit Thomas Job (20128007)
Siddhant Patny (20128102)
Yatyendra Singh (20128068)
Aditya Kumar Gaur (20129024)

to the
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD
May, 2016

UNDERTAKING

We declare that the work presented in this project titled "*Twitter Trends*", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Allahabad, for the award of the ***Bachelor of Technology*** degree in ***Information Technology***, is our original work. We have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, we accept that our degree may be unconditionally withdrawn.

May, 2016
Allahabad

Medha Srivastava
Rohit Thomas Job
Siddhant Patny
Yatyendra Singh
Aditya Kumar Gaur

CERTIFICATE

Certified that the work contained in the project titled "*Twitter Trends*", by *Group 1 (B.Tech 8th Semester, IT)*, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Professor M. M. Gore)
Computer Science and Engineering Dept.
M.N.N.I.T, Allahabad

May, 2016

Preface

The age of the Internet has seen globalization and the interaction of cultures like never before. These social networking sites like Twitter have provided a platform to understand the ideologies and communicate with people from all around the world. As human beings, it is in our nature to want to stay updated with the daily news and trends from around the world. To understand and keep track of this myriad of information shared by people we present Twitter Trends, a web-based application which automatically extracts and processes trending topics from all over the world and provides the user a platform to get topic world cloud, its trends graph, live tweets and related topics for a particular trending topic in interest of the user.

Acknowledgements

Apart from our efforts, the success of any work depends largely on the encouragement and guidance of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We feel a heartiest sense of obligation to **Prof. M. M. Gore**, Professor of Computer Science for providing invaluable support and guidance throughout the project. He imparted great help and technical guidance at each and every step of the project to make it a success. The efforts put by us would have not been fruitful, if it were not for the people around us, who encouraged us at all times.

We also express our sincere thanks to all our professors of Computer Science and Engineering Department, who helped us in laying the strong foundation for topics such as DBMS, Data Mining and Technical Writing, which constitutes the knowledge base for this project.

List of Figures

Contents

Chapter 1

Introduction

In the last decade, social media activity have reached enormous levels. Hundreds of millions of users now participate in online social networks and forums, subscribe to microblogging services or maintain blogs. Twitter, in particular, is currently the major microblogging service, with more than 11 million active subscribers.

Twitter users post 140 characters long messages called *tweets*, to report their current thoughts and actions, comment on breaking news and engage in discussions. These messages contain hashtags which are short phrases that convey a particular messages. The product of social activity on Twitter reaches an estimated total of over 500M tweets per day[3]. Every tweet is associated with an explicit timestamp that declares the exact time it was generated. Moreover, every user has a well-defined profile with personal information (name, location, biographical sketch). Such a document stream contains a great wealth of information and offers significant opportunities for exploration, as well as challenges.

One of the primary challenges which we try to address with our system, is to automatically detect and analyze the emerging topics through hashtags and user mentions (i.e. the trends) that appear in the stream, and do so in real time. Trends are typically driven by emerging events, breaking news and general topics that convey what the what the world is talking about and attract the attention of a large fraction of Twitter users[6].

1.1 Motivation

It is a general tendency among humans to stay updated with the current events and issues in the world. Trend detection is of high value to news reporters and analysts, as they might point to fast-evolving news stories. Trend detection is also important for online marketing professionals and opinion tracking companies, as trends point to topics that capture the publics attention. The requirement for real-time trend detection is only natural for a live stream where topics of discussion shift dynamically with time. Furthermore, for such a system to be scalable over massive document streams, an approach is required that makes as few passes over the data as possible[5][7]. This had made it a hot topic of research among the academia on especially pertaining to the real-time analysis of tweets ([2][3]).

1.2 Applications

Trend detection has a variety of applications. Provided the accuracy of the models used to detect trends and learn from them is high enough, the information gained can lead to groundbreaking results. Some of the applications that we had in mind include:

- **Social Analysis:** Analysis of interests by region and its lifespan. Any kind of social analysis can help us gain an insight into the current psychology of the people. This can help us determine how the public will react to some event and make necessary preparations in advance.
- **Finance:** Information related to companies and business magnates tend to surface on social networks like Twitter even before the official press release. These kinds of information can be useful in determining the stock trends and ultimately even to predict stock prices.
- **Marketing:** People often share their opinions on anything and everything. This can be used to determine the interests of an individual or group. Based on

these interests products can be targeted to specific audiences in order to maximize profit.

- **News Tracking:** Information surfaces on Twitter very quickly. This allows news and the updates regarding it to be available instantaneously. Since Twitter is a global platform, even international news can be updated in real-time.

Thus extracting trends from Twitter can prove to be extremely useful. In what follows, we describe how our product tackles the challenge of real-time trend detection, as well as its architecture. We conclude with a description of our demonstration scenario.

1.3 Work Done in the Previous Semester

We had previously outlined methods to extract live streaming tweets and tracking real-time trends. We had presented a novel method to automate the processing and trend extraction along with an optimized method to preprocess the tweets using the MapReduce model and store it in a NoSQL database. This had enhanced the speed of computing the vast amount of tweets and created a framework on which further analysis can be done. The current state of the project allows room for extensions in the form of interfacing with other projects applying natural language processing techniques to Twitter. Finally, once all results are analyzed, this project hopefully will have demonstrated the ability of natural language processing tools to extract and identify pertinent information from this live stream of unorganized data.

Chapter 2

Technical Specifications

2.1 Programming Languages Used

1. **Python:** Python is a widely used general-purpose, high-level programming language. Python is a very powerful language with a wide range of packages that can be easily imported into any project. The reasons why we decided to use **Python** as our **primary** programming language for this project are as follows:

   (a) Python has a very small learning curve.

   (b) Python provides great readability which makes it easier to debug.

   (c) Python offers *Numpy* and *Scipy* packages that provide easy access to machine learning tools.

   (d) Python is compatible with all the databases both Relational and Non-Structured.

2. **Windows Batch Scripting:** A batch file is a kind of script file in DOS and Windows. It consists of a series of commands to be executed by the command line interpreter, stored in a plain text file. As we have used Windows as our deployment Operating System, batch scripting was essential in order to automatically run necessary processes before initialising the software.

3. **JavaScript:** Javascript was used to program the map and reduce functions for aggregation. MapReduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. A MapReduce program is composed of a Map() procedure (method) that performs filtering and sorting (such as sorting tweets by first name into queues, one queue for each name) and a Reduce() method that performs a summary operation (such as counting the number of tweets in each queue, yielding tweet frequencies).

4. **Web Development Languages:** HTML, JavaScript, CSS and Python were the languages used to create and maintain the web portal.

2.2 Tools Used

1. **Django:** Django is a python based web framework for web development[9]. Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

2. **MongoDB:** MongoDB is a cross-platform document-oriented database. Classified as a NoSQL database, MongoDB eschews the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas (MongoDB calls the format BSON), making the integration of data in certain types of applications easier and faster.
   We have used MongoDB to store all raw tweets and aggregates. The data structures used by NoSQL databases (e.g. key-value, wide column, graph, or document) differ slightly from those used by default in relational databases,

making some operations faster in NoSQL. As NoSQL databases are increasingly being used in big data and real-time web applications, we have prefered this DBMS over MySQL for our Mining Engine.

3. **Bootstrap:** Bootstrap is a free and open-source collection of tools for creating websites and web applications [10]. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. It aims to ease the development of dynamic websites and web applications.

2.3 Software Interfaces

1. **Twitter Streaming API:** API provided by Twitter that gives the product access to 1% of the latest tweets which can be filtered according to given key words[8]. It uses a persistent HTTP connection open, for retrieving the data.

2. **Twitter Search API:** API provided by Twitter that gives product access to Twitters search results, focussing more on relevance than completeness. Twitter Search API was used to fetch live tweets corresponding to each trending topic.

Chapter 3

Architecture

3.1 Introduction

In this chapter we present the architecture of our system shown in Figure 1, required to extract, process the tweets and display the relevant results. We provide detailed explanations for each component as well as its technical specifications and advantages.
Our system is divided into two main sections:

- Mining Engine: The main backend system that performs the data mining and processing.
- Web Portal: The user interface that enables the user to interact with the results of the engine.

3.2 Mining Engine

The Mining Engine is responsible for extracting tweets using the Twitter Streaming API, process them, store them into a MongoDB collection for efficient retrieval, perform aggregation and analysis and integrate the results with the web portal using an SQLite database.
We now describe each of the individual functions in detail.

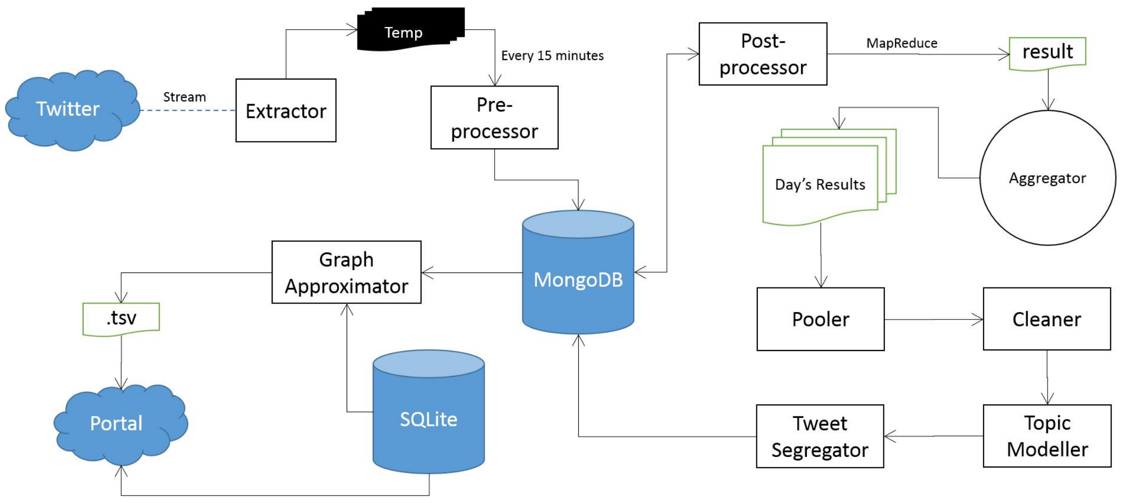


Figure 1: Flowchart depicting the architecture

3.2.1 Tweet Extraction

A process runs in the background which uses the Twitter Streaming API to continuously extract tweets. The extraction process runs each day from 00:05 to 23:55 local time. The ten minute gap prevents overlap between two days. The Streaming API provides a stream of tweets according to filters provided by the user. The keywords used to filter tweets are the top ten words of the most frequent words used on Twitter[11] and English is the language filter. Since Twitter only provides 1% of the total tweets filtered, Twitter provides the Twitter IDs of the missed tweets as part of the stream. The tweet extractor handles such messages, retrieves the tweets in JSON format, and stores them in a flat file. Once 10,000 tweets have been written to the file, the file is moved into a temporary directory for processing.

3.2.2 Pre-processing

A raw tweet contains lot of information about the tweet that are not necessary. For efficient storage and processing, the tweet is pre-processed before dumping it into MongoDB. The pre-processing scripts loads the JSON files from the temporary directory into memory in the form of python dictionaries. The information retained includes the tweet ID, timestamp, the actual tweet, URLs mentioned in the tweet, location of origin, number of retweets and the user who created the tweet. Stop words are removed using the list of stop words from python 'nltk' package. Only those tweets that contain entities such as hashtags and user mentions are dumped into MongoDB, the rest are discarded. These pre-processed tweets are then dumped into a temporary raw collection. To efficiently dump the data, bulk writes are performed, where each write dumps 10,000 documents into the collection.

3.2.3 Entity Aggregation

Entity aggregation is done on all the tweets collected in a day. To save time, this entity aggregation is done at regular intervals. The entity aggregation script runs just after the pre-processor dumps data into the temporary raw collection. MapReduce is done on the temporary collection to extract frequency of the entities and outputted into a temporary results collection. To merge the new data, MapReduce is done on the temporary results collection and the conflicting documents of the days results collection is reduced using the same reduce function. Post this, the data in the temporary raw collection is dumped into the days raw collection and both the temporary collections are dropped from the database. The pre-processing and entity aggregation is repeated every 15-20 minutes.

3.2.4 Topic Modelling and Trends Extraction

Latent Dirichlet Allocation is perfomed to extract the topics of the document of tweets. It assumes that documents (assumed to be bags of words) are generated by a mixture of topics (distributions over words). In our case, the LDA algorithm is used to process a set of documents(collections in database) of tweets prepared after the extraction and pooling process. LDA provides us with a list of 'n' topics that constitute the documents. These topics are represented by a group of weighted words which can also be represented as a word cloud. Therefore, each word cloud can be considered as a trending topic.
To improve the efficiency of the process, tweets containing the same entity are

pooled together in the same document. The top 100 entities are considered for pooling.

3.2.5 Entity Categorization

Once the relevant words describing a particular topic are extracted, the entities in the tweet are assigned to the relevant collection based on the best match to assign the topic for that tweet. Since a tweet can have multiple entities, it can belong to multiple collections. Thus we can categorize our stream of tweets into its respective category and information can then be mined for each category. Classification is performed on the top 100 documents.

3.2.6 Graph Approximation

For each entity, the data for the number of tweets against timeline has to be plotted for trend analysis. For this purpose, number of tweets integrated for intervals of one hour is computed, and stored in a TSV file. Taking this data as input, a graph is plotted for the trend analysis of the topic. The number of tweets is computed from the raw collection. The raw collection uses a compound index [entities:ASCENDING, timestamp:ASCENDING]. This index makes the query for each time interval execute in O(log n) time. The timestamp stored in the raw collection is a python datetime object (UTC). The local time is automatically converted to UTC before making the query. For each entity in the SQLite database, 24 queries are made to the database for each time interval. This may seem inefficient, but as the number of entities is 10, the computation is faster than MapReduce. The data so obtained is stored in the portals directory, one TSV file for each entity.

3.2.7 Relevant Article Extraction

Once the entity categorization is done, a MapReduce is performed on each of the new collections to find the frequency of shared URLs. Once the most frequent URLs for each topic is found, we crawl the corresponding webpage. A content extraction algorithm is run on the HTML to extract any relevant article from the web page. If the article is suitable, we then display it on the portal.

3.2.8 Automation

The engine is completely automated so that there is no need for any supervision. The use of subprocess and multiprocess libraries, enabled the spawning of new processes for each of the above functions so as to make use of the multiple processors available to us. The time synchronisation is obtained using accurate alarms that are deployed at strategic places. The processes that are force killed go through a cleaning process, where all its children are killed and resources released.

3.3 Web Portal

The web portal acts as the UI for displaying the results of the mining engine. Following features have been incorporated into it:

- **Top trending hashtags and mentioned users:** These are the entities extracted from the SQLite database. They show the top 10 trending hashtags and user mentions. Each entry has a detailed graphs and live streaming tweets linked to it.

- **Trend Graph:** Each topic has an associated graph that shows the trend pattern in the time domain. The graph is generated using SVG and D3 JavaScript library. The data points are obtained from the TSV file generated by the Graph Approximator.

- **Live Tweets related to the topic:** Each entity has its own dedicated web page. This page contains the graph as stated above and the live tweets. The live tweets are extracted directly from Twitter using its Search API.
  The web page makes AJAX requests to the server containing the keyword

to be searched (i.e. the hastags/user mentions) and the latest tweet ID displayed. Once each request is retrieved, the server makes a request for the latest tweet with an ID greater than the one given and which contains the given keyword in its text. Note that the keyword search is case-insensitive.

- **Topic Related Information:** Topic related tweets and URLs are displayed for each category once the tweets are classified. This is extended to extract the most frequently shared URLs in the tweets belonging to that category as well as the main content of articles which is then displayed alongside the graph and live tweets.

3.4 Summary

We will now summarize the entire architecture and its workflow as shown in Figure 1. Control starts with the engine manager. The engine manager calls the extractor and processor. The extractor extracts tweets using the Streaming API and stores it in a flat file. The processor reads the flat files and extracts relevant information and dumps it into a MongoDB temporary collection. The daily aggregator, also part of the processor, computes the aggregate and integrates it into the day's result collection. It then persists the temporary collection. The processor executes every 15 minutes.

At the end of the day, an 'alarm' rings post which the extractor and processor is terminated. Then the topic modeller extracts the top entities as required. Finally, the graph approximator generates data points for the graph of each entity using the collections of each topic. This data is then dumped into a TSV file which can be accessed by the web portal. Then all the URLs are aggregated to find the most frequently shared URLs. After that the respective articles are crawled and content is extracted.

The extractor and processor is then restarted to collect and process data for the next day. These were the integral modules of our system. We now go on to describe our results and the challenges we faced while designing these features.

Chapter 4

Results and Challenges

4.1 Results

We performed the analysis from 11/04/2016 14:00 to 12/04/2016 14:00. All the tweets were then modelled to extract topics and segregated according to the topic distribution. The relevant graph for one of the topics for the last week is shown in Figure 2. For each topic we also extracted and displayed a relevant article using normal frequency analysis and a good content extraction algorithm. The corresponding article is shown in Figure 3.

4.2 Challenges

- **Restrictions from Twitter:** Twitter only provides 1% of the actual volume of tweets generated matching the query made by the extractor. Even though the number of tweets received is adequate, the guarantee of it being representative is still a problem to be addressed.

- **Storage:** Twitter provides a lot of information about the tweets through its Streaming API. Around 3 Million tweets are downloaded per day which takes over 15 GB of memory only for raw text. If we are to store images associated with each tweet, the memory constraints would be violated. In order to optimize memory usage, we extracted relevant data as it came and

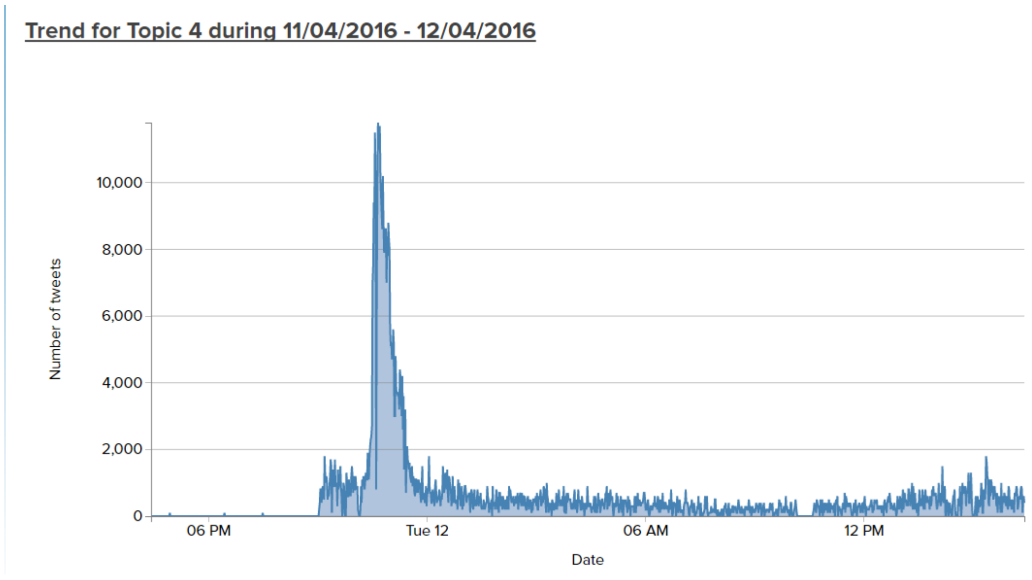


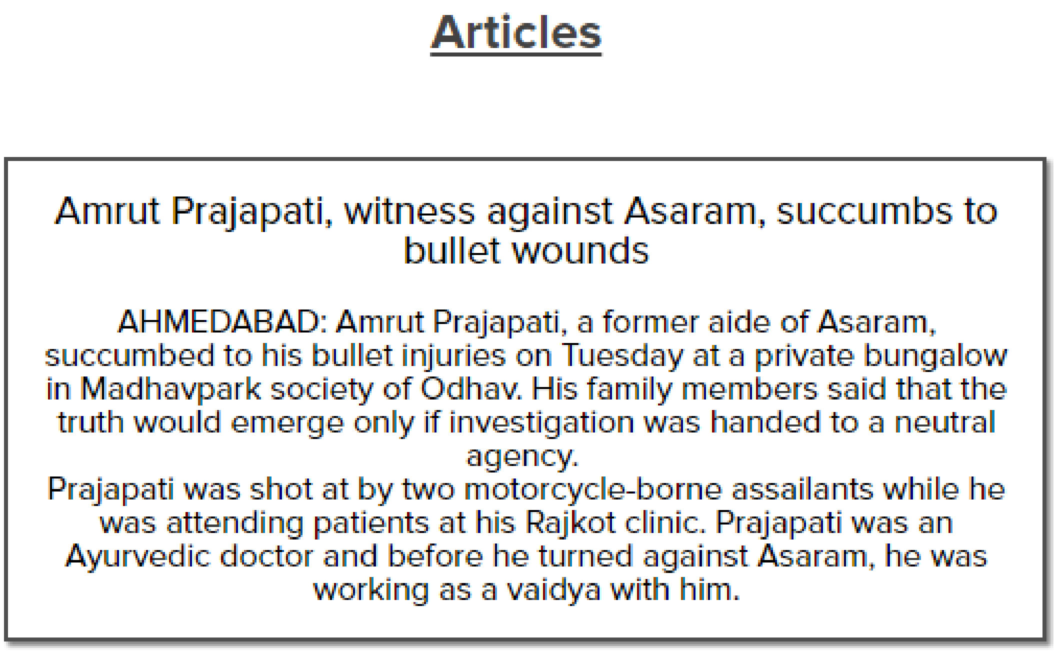
Figure 2: Trend Graph for Topic related to Asaram

deleted the original flat files. Also instead of actually storing the images, we only stored references to it (URLs), when available.

- **Processing of the vast amount of tweets:** Our MapReduce model was able to enhance the speed for a limited number of tweets but distributed processing would be required for analysis over a long time-frame.

- **Historical Data:** For making any kind of predictions, we will have to first train our model. For this purpose, we need historical data which is accurate and representative. For better predictions, the amount of data also matters which might be a problem given our storage constraints.

- **Reliable Internet Connection:** For continuous data extraction, the underlying Internet connection should be reliable. Even if access to the Internet is always available, the bandwidth provided has to be more or less constant for any accurate scientific calculations. If the bandwidth cannot be maintained, any normalization techniques have to be deployed.



Articles

Amrut Prajapati, witness against Asaram, succumbs to bullet wounds

AHMEDABAD: Amrut Prajapati, a former aide of Asaram, succumbed to his bullet injuries on Tuesday at a private bungalow in Madhavpark society of Odhav. His family members said that the truth would emerge only if investigation was handed to a neutral agency.
Prajapati was shot at by two motorcycle-borne assailants while he was attending patients at his Rajkot clinic. Prajapati was an Ayurvedic doctor and before he turned against Asaram, he was working as a vaidya with him.

Figure 3: Article related to Topic

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Twitter being the major microblogging service is a reliable source for trends detection. We have extracted live streaming tweets, processed them to find top hashtags and user mentions and displayed details for each trending topic using trends graph, live tweets and summary of related articles. This data can be used to support social analysis, finance, marketing or news tracking. Once we had extracted the live streaming tweets and created an automated framework for anaylsis, we implemented Topic Modelling and Entity Categorization to classify the tweets and extract valuable information about its contents and find similiar tweets and related articles and URLs. Thus we have the required information to get an overhaul of the topics which are trending at that particular time.

5.2 Future Work

To expand our framework and make it more intuitive by providing extensive analysis and customizations to the user, there are several extensions that can be implemented using the relevant Data Mining and Language Processing tools[4]. The preferences of the user can be taken into account by adding live tweets based on the interests of the user as well as the ability to search for a particular trend. Our model can also be extended to perform social analysis by learning the inherent characteristics and variance of a trend which can be used to predict its variance with time in the future as well as to estimate future trends. Once these features are incorporated we will have an intelligent system for continuous real-time trend analysis to be used for a foray of applications.

References

[1] Mori, Masaki, Takao Miura, and Isamu Shioya. *Topic detection and tracking for news web pages.* Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. IEEE Computer Society, 2006.

[2] Lu, Rong, and Qing Yang. *Trend Analysis of News Topics on Twitter.* International Journal of Machine Learning and Computing 2.3 (2012): 327.

[3] Benhardus, James, and Jugal Kalita. *Streaming trend detection in twitter.* International Journal of Web Based Communities 9.1 (2013): 122-139.

[4] Kumar, Shamanth, Fred Morstatter, and Huan Liu. *Twitter data analytics.*. Springer, 2014.

[5] Mathioudakis, Michael, and Nick Koudas. *Twittermonitor: trend detection over the twitter stream.* Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM, 2010.

[6] Java, Akshay, et al. *Why we twitter: understanding microblogging usage and communities.* Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis. ACM, 2007.

[7] Bansal, Nilesh, and Nick Koudas. *Blogscope: a system for online analysis of high volume text streams.* Proceedings of the 33rd international conference on Very large data bases. VLDB Endowment, 2007.

[8] Twitter Developers,
`https://dev.twitter.com/`

[9] Django Web Framework,
`https://www.djangoproject.com/`

[10] Bootstrap Front-end Framework,
`http://getbootstrap.com/`

[11] Most Used Words in Twitter Survey
`http://techland.time.com/2009/06/08/the-500-most-frequently-used-words-on-twitte`