International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 05 | May -2017 www.irjet.net p-ISSN: 2395-0072

Tweet Analysis for Real-Time Event Detection and Earthquake Reporting System Development

Ekta, Paahuni Khandelwal, Priya Bundela, Richa Dewan

[1,2,3,4]*Student, Dept. Of Computer Science, Miranda House (University of Delhi), Delhi, India*

Abstract - *Twitter is one of the most popular online social networking sites. An important characteristic of Twitter is its real-time nature. We investigate the real-time interaction of events such as earthquakes in Twitter and propose an algorithm to monitor tweets and to detect a target event. To detect a target event, we devise a classifier of tweets based on features such as the keywords in a tweet, the number of words, and their context. Subsequently, we produce a probabilistic model for the target event occurrence. We regard each Twitter user as a sensor. Because of the numerous earthquakes and the large number of Twitter users across, we can detect an earthquake with high probability merely by monitoring tweets.*

***Key Words*: Twitter, event detection, social sensor, location estimation, earthquake**

1. INTRODUCTION

TWITTER, a popular micro blogging service, has received much attention recently. This online social network is used by millions of people around the world to remain socially connected to their friends, family members, and co-workers through their computers and mobile phones. Twitter asks one question, "What's happening?" Answers must be fewer than 140 characters. A status update message, called a tweet, is often used as a message to friends and colleagues. A user can follow other users; that user's followers can read her tweets on a regular basis. A user who is being followed by another user need not necessarily reciprocate by following them back, which renders the links of the network as directed. Since its launch on July 2006, Twitter users have increased rapidly. The number of registered Twitter users exceeded 100 million in April 2012. The service is still adding about 300,000 users per day. Currently, 190 million users use Twitter per month, generating 65 million tweets per day. Many researchers have published their studies of Twitter to date, especially during the past year. Most studies can be classified into one of three groups: first, some researchers have sought to analyze the network structure of Twitter. Second, some researchers have specifically examined characteristics of Twitter as a social medium. Third, some researchers and developers have tried to create new applications using Twitter. Twitter is categorized as a micro blogging service. Micro blogging is a form of blogging that enables users to send brief text updates or micro media such as photographs or audio clips. Micro blogging services other than Twitter include Tumblr, Plurk, Jaiku, identi.ca, and others [1]. Our study, which is based on the real-time nature of one social networking service, is applicable to other micro blogging services, but we specifically examine Twitter in this study because of its popularity and data volume. An important characteristic that is common among micro blogging services is their real-time nature [2]. Although blog users typically update their blogs once every several days, Twitter users write tweets several times in a single day. Users can know how other users are doing and often what they are thinking about now, users repeatedly return to the site and check to see what other people are doing. Several important instances exemplify their real-time nature: in the case of an extremely strong earthquake in Nepal, many pictures were transmitted through Twitter. People were thereby able to know the circumstances of damage in Nepal immediately. In another instance, when an airplane crash-landed on the Hudson River in New York, the first reports were published through Twitter and Tumblr. In such a manner, numerous update results in numerous reports related to events. They include social events such as parties, baseball games, and presidential campaigns. They also include disastrous events such as storms, fires, traffic jams, riots, heavy rainfall, and earthquakes. Actually, Twitter is used for various real-time notifications such as that necessary for help during a large-scale fire emergency or live traffic updates. An editor in Chief at Mashable, a social media news blog, wrote in his blog about the interesting phenomenon of real-time media:

Earthquake Shakes Twitter Users ... And Beyonce: Earthquakes are one thing you can bet on being covered on Twitter first, because, quite frankly, if the ground is shaking, you're going to tweet about it before it even registers with the USGS and long before it gets reported by the media. That seems to be the case again today, as the third earthquake in a week has hit the country and its surrounding islands, about an hour ago. The first user we can find that tweeted about it was Ricardo Duran of Scottsdale, AZ, who, judging from his Twitter feed, has been travelling the world, arriving in the country yesterday.

This post well represents the motivation of our study. The research question of our study is, "can we detect such event occurrence in real-time by monitoring tweets? "This paper presents an investigation of the real-time nature of Twitter that is designed to ascertain whether we can extract valid information from it. In this research, we take the following

steps: first, we extract the twitter data; next, we crawl through numerous tweets related to target events; next, we use classifying and filtering algorithms to filter out the relevant tweets and lastly, we propose probabilistic models to extract events from those target tweets. Here, we explain our methods using an earthquake as a target event. First, to obtain tweets on the target event precisely, we apply semantic analysis of a tweet. For example, users might make tweets such as "Earthquake!" or "Now it is shaking," for which earthquake or shaking could be keywords, but users might also make tweets such as "I am attending an Earthquake Conference," or "Someone is shaking hands with my boss." We prepare the training data and devise a classifier using Naïve Bayes algorithm. After doing so, we obtain a probabilistic model of an event. We then make a crucial assumption: each Twitter user is regarded as a sensor and each tweet as sensory information. These virtual sensors, which we designate as social sensors, are of a huge variety and have various characteristics: some sensors are very active; others are not. A sensor might be inoperable or malfunctioning sometimes, as when a user is sleeping, or busy doing something else. Consequently, social sensors are very noisy compared to ordinary physical sensors. Regarding each Twitter user as a sensor, the event-detection problem can be reduced to one of object detection and location estimation in a ubiquitous/ pervasive computing environment in which we have numerous location sensors: a user has a mobile device or an active badge in an environment where sensors are placed.

The contributions of this paper are summarized as follows: The paper provides an example of integration of semantic analysis and real-time nature of Twitter, and presents potential uses for Twitter data.

For earthquake prediction and early warning, many studies have been made in the seismology field. This paper presents an innovative social approach that has not been reported before in the literature.

This paper is organized as described below. In the next section, we explain an investigation of Twitter users and earthquakes in the real world. The next Section presents our explanation of semantic analysis and sensory information with subsequent the probabilistic model in the following Section. After that, we describe the experiments and evaluation of event detection, Twitter APIs, data extraction, data filtration. Finally, we conclude the paper.

2. INVESTIGATION

Our study involves two investigation phases, namely, Primary and Secondary.

2.1 Primary

We choose earthquakes as target events, based on the preliminary investigations. We explain them in this section. First, we choose earthquakes as target events for the following reasons:

a) Seismic observations are conducted worldwide, which facilitates acquisition of earthquake information, which also makes it easy to validate the accuracy of our event detection methodology; and

b) It is quite meaningful and valuable to detect earthquakes in earthquake-prone regions.

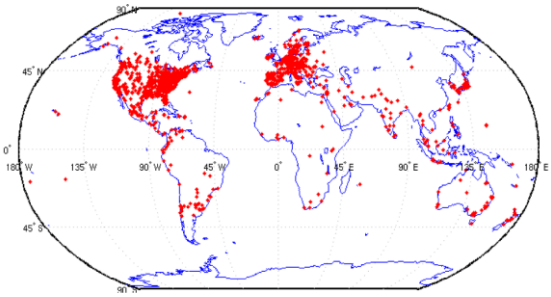

Fig - 1: Twitter User Map

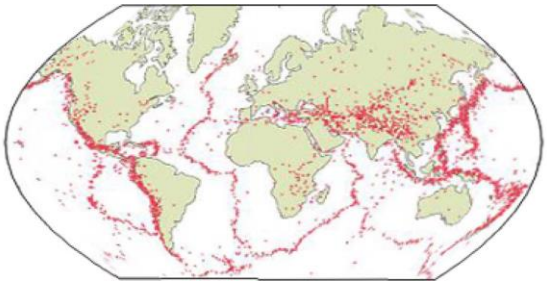

Fig - 2: Earthquake Map

Fig. 1 portrays a map of Twitter users worldwide (obtained from UMBC eBiquity Research Group); Fig. 2 depicts a map of earthquake occurrences worldwide. Comparing the two, we observed that as there are numerous earthquakes occurring worldwide, there are equally numerous amount of social sensors that are distributed throughout the world. Therefore, it is becomes possible to detect an earthquake by monitoring tweets.

2.2 SECONDARY

In Secondary research, a survey was conducted on Twitter usage as a Social Media among Indian people for an analysis which helped us gain an insight into India's current Twitter community, which we used to help outline and define our target personas. We performed the Google survey by circulating Google form link in order to reach as many people as possible. The analysis opened additional insight into the type of people the Twitter marketing is attracting and helped us further in defining their profiles with information such as; what they are interested in, number of followers and following they have and so on.

Among these users, 45.8 % users log on to their account more than once in a day. Majority of them use Twitter platform to keep up with the news in general (63.7 %) and because they want a place where they can post things they want to say immediately (29.4 %).

3. EVENT DETECTION

As described, we target event detection. An event is an arbitrary classification of a space-time region. An event might have actively participating agents, passive factors, products, and a location in space/time [3]. We target events such as earthquakes, typhoons, and traffic jams, which are readily apparent upon examination of tweets. These events have several properties.

a) They are of large scale (many users experience the event).
b) They particularly influence the daily life of many people (for that reason, people are induced to tweet about it).
c) They have both spatial and temporal regions (so that real-time location estimation is possible).

Such events include social events such as large parties, sports events, exhibitions, accidents, and political campaigns. They also include natural events such as storms, heavy rains, tornadoes, typhoons/hurricanes/cyclones, and earthquakes. We designate an event we would like to detect using Twitter as a target event.
In this section, we explain how to detect a target event using Twitter. First, we crawl tweets including keywords related to a target event. From them, we extract tweets that certainly refer to a target event by creating algorithms for the same. Second, we detect a target event by using a probabilistic approach.

3.1 TWEET ANALYSIS

To detect a target event from Twitter, we search from Twitter and find useful tweets. Our method of acquiring useful tweets for target event detection is portrayed. Tweets might include mention of the target event. For example, users might make tweets such as "Earthquake!" or "Now it is shaking." Consequently, earthquake or shaking might be keywords (which we call query words). However, users might also make tweets such as "I am attending an Earthquake Conference." or "Someone is shaking hands with my boss." Moreover, even if a tweet is referring to the target event, it might not be appropriate as an event report. For instance, a user makes tweets such as "The earthquake yesterday was scary." or "Three earthquakes in four days." These tweets are truly descriptions of the target event, but they are not real-time reports of the events. Therefore, it is necessary to clarify that a tweet is truly referring to an actual contemporaneous earthquake occurrence, which is denoted as a positive class. To classify a tweet as a positive class or a negative class, we use Naïve Bayes approach, which is a widely used machine-learning algorithm. By preparing positive and negative examples as a training set, we can produce a model to classify tweets automatically into positive and negative categories.

3.2 TWEET AS A SENSORY VALUE

We can search the tweet and classify it into a positive class if a user makes a tweet about a target event. In other words, the user functions as a sensor of the event. If she makes a tweet about an earthquake occurrence, then it can be considered that she, as an "earthquake sensor," returns a positive value. A tweet can therefore be regarded as a sensor reading [4]. This crucial assumption enables application of various methods related to sensory information.

Assumption 1. Each Twitter user is regarded as a sensor. A sensor detects a target event and makes a report probabilistically.

Observation by sensors corresponds to an observation by Twitter users. They are converted into values using a classifier. The virtual sensors (or social sensors) have various characteristics: some sensors are activated (i.e., make tweets) only by specific events, although others are activated by a wider range of events. The sensors are vastly numerous: there are more than 100 million 'Twitter sensors' worldwide producing tweet information around the clock. A sensor might be inoperable or operating incorrectly sometimes (which means a user is not online, sleeping, or is busy doing something else). For that reason, this social sensor is noisier than ordinary physical sensors such as location sensors, thermal sensors, and motion sensors. Therefore, a probabilistic model is necessary to detect an event.

Assumption 2. Each tweet is associated with a time and location, which is a set of latitude and longitude coordinates.

By regarding a tweet as a sensory value associated with location information, the event detection problem is reduced to detection of an object and its location based on sensor readings. Estimating an object's location is arguably the most fundamental sensing task in many ubiquitous and pervasive computing scenarios. In this research field, some probabilistic models are proposed to detect events by dealing appropriately with sensor readings.

4. MODEL

Our study uses the probabilistic model for event occurrence using Naïve Bayes algorithm.

4.1 NAÏVE BAYES ALGORITHM

The Model that we are proposing is based on the Naive Bayes algorithm. It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'. Naïve Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

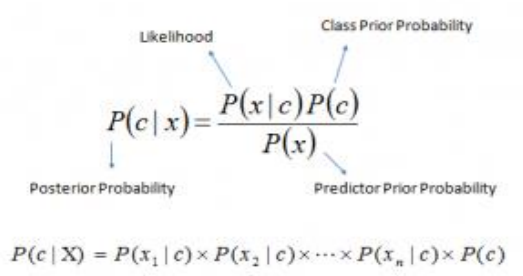

Fig – 3: Bayes Theorem

Above,

a) $P(c|x)$ is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
b) $P(c)$ is the prior probability of *class*.
c) $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
d) $P(x)$ is the prior probability of *predictor*.

Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time [5]. Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)

4.2 EVENT OCCURRING PROBABILITY

After classifying a tweet into positive or negative case, we can calculate the event occurrence probability. We form a probabilistic model that gives the probability of event occurrence, for a given tweet that is a positive example.
If the probability is larger than a predetermined threshold (based on the past research), then it determines an actual occurrence of the target event. To assess an alarm, we must calculate the reliability of multiple sensor values. For example, a user might make a false alarm by writing a tweet. It is also possible that the classifier misclassifies a tweet into a positive class.

4.3 POSTERIOR PROBABILITY

The module approaches as:

a) First the words of length < 3 are removed from consideration such as 'a', 'an', 'I', 'hi', 'is', 'Oh', etc. (as they majorly do not affect the tweet).
b) Using Counter function calculates the frequency of each word in the tweet.
c) Using this frequency and the total denominator of all the words and Naïve Bayes, posterior probabilities are calculated for both negative and positive class.
d) These probabilities are then written onto text files namely posterior_pos.txt and posterior_neg.txt.

4.4 PROBABILITY ESTIMATOR MODULE

a) First takes the input from the user.
b) Perform filter functions to remove #. @, links, images etc. from the entered tweet.
c) Words of length < 3 are removed from the entered tweet such as 'a', 'an', 'I', 'hi', 'is', 'Oh', etc. (as they majorly do not affect the tweet and are also not present in the dataset considered).
d) Then it splits the tweet into individual words
e) Those words are then searched throughout the dataset and their posterior probabilities are taken into consideration for calculating net probability as

P (net probability | word) = P (positive | word) - P (negative | word) (1)

f) The words that are not present in the tweet are simply discarded.
g) The net probabilities of all the words present in the tweet are added and then their sum is divided by the number of words present in the tweet to normalize the result.

Sum = P (net probability | word1) + P (net | probability | word2) +P (net… | wordn) P (estimated) =Sum / word count (2)

h) This gives the estimated probability of a new tweet and is displayed as output.

5. DATA COLLECTION AND EVENT TAGGING

5.1 TWITTER APIs

To achieve our goal of real-time analysis for event recognition, we retrieve as many relevant tweets as fast as possible and from as many users as possible. Twitter provides three application programming interfaces (APIs), i.e., REST, Search and Streaming API. Fig.3 shows the database structure and the APIs. In the figure, the arrows on the right represent APIs.

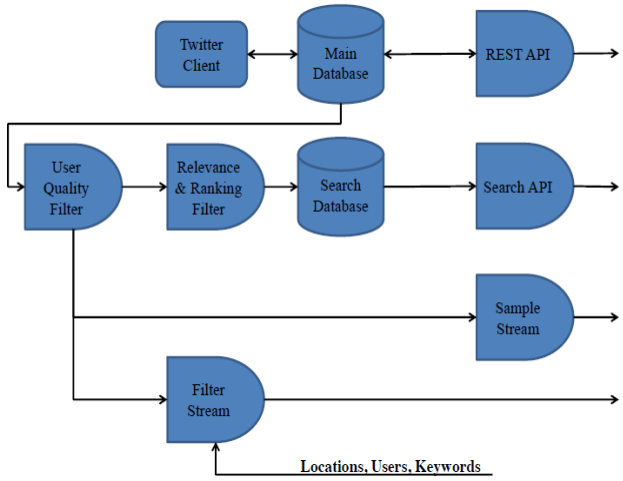

Fig – 4: Twitter APIs and database structure. The top arrow on the right represents REST API, the second arrow represents Search API, and the bottom two represent the Streaming API.

The Representational State Transfer (REST) API allows developers to access core Twitter data stored in the main database, which contains all the tweets. Through the REST API, developers can retrieve Twitter data including user information and chronological tweets [6]. For example, the *home timeline* includes the 20 most recent tweets on a user's home page; the *public timeline* returns the 20 most recent tweets in every 60 seconds. These limitations make the REST API not particularly suitable for real-time tweet collection; the REST API is best for collecting a large number of tweets from specific user IDs off-line.

The Search API will return tweets that match a specified query; however it will only search a limited subset of tweets posted in past 7 days in the main database. The query parameters include time, location, language etc. Twitter limits the return results to 100 tweets per request. Although the Search API is able to collect tweets in real-time, one cannot control the topic of the returned tweets. Twitter limits the request rate to the REST and Search API to 150 per hour by default. It previously allowed up to 20,000 search requests per hour from white-listed IPs (about six requests per second), but, unfortunately, Twitter no longer grants white-listing requests since Feb 2011. This limitation makes the REST and Search APIs unsuitable for real-time event detection.

The Streaming API offers near real-time access to Tweets in sampled and filtered forms. The filtered method returns public tweets that match one or more filter predicates, including *follow*, *track*, and *location*, corresponding to user ID, keyword and location, respectively. Twitter applies a User Quality Filter to remove low quality tweets such as spams from the Streaming API. The quality of service of the Streaming API is best-effort, unordered and generally at-least-once; the latency from tweet creation to delivery on the API is usually within one second. However, reasonably focused track and location predicates will return all occurrences in the full stream of public statuses. We have used a streaming API in our study to extract twitter data for the following reasons:

a) All the tweets returned are up to date;
b) There is no rate limit; and
c) The track filter predicate allows us to collect tweets that are related to the target event using keywords.

Although there is no explicit rate limit, we cannot obtain all public tweets and we report our observation of an undocumented restriction in the Streaming API.

5.2 TARGET EVENTS

We based our study on detecting earthquakes around the world. Hence, we collected twitter data, keeping our target event as 'earthquakes'.
For this purpose, we collected earthquake related tweets using the Streaming API using specific target keywords. We collected the tweets and their metadata such as tweet source, created time, location, and device. These tweets were analyzed for event recognition through various algorithms created. A large volume of tweets were collected to draw out the best of results on event occurrence of an earthquake.

5.3 PREPARING DATASET

To get positive and negative tweet dataset, we collected earthquake related tweets using the Streaming API. We also used a Python library called **Tweepy** to connect to Twitter Streaming API and to download the data. Tweepy is open-sourced, hosted on GitHub and enables Python to communicate with Twitter platform and use its API. Tweepy supports accessing Twitter via the method, OAuth. Tweepy is probably the best Twitter library for Python, especially when considering the Streaming API support [7].

Positive tweets were collected by taking target event as Earthquake and Date. Here date value is actual date when the earthquake occurred in past.
Similarly negative tweets were collected by taking target event Earthquake on day when there when earthquake didn't occurred. The search result gave tweets having keyword earthquake, however since no earthquake occurred that day, so all those tweets were considered negative ones.

5.4 EXTRACTING TEXT FROM TWEETS DATASET

Tweet structures comprise of metadata, such as id, created on, text, location, entities and so on. Our actual tweet data in under "text" attribute. So to extra this text value, both positive and negative tweets files are searched for keyword "text" and characters are printed in text file.

5.5 FILTERING TWEETS

The fourth module **Tweet filter** is responsible for removing the unnecessary characters like punctuation marks (for smileys, hashtags, etc.) in a tweet that are not required in calculating the probability of the tweet further [8]. This module takes input as text files which are generated as output of previous module and generates further two files as output containing the modified tweets.

6. CONCLUSION

Real time event detection system aims at detecting some events in real time and take the required actions based on the detection. Our project **"Real time event detection of earthquake using twitter "** aimed at building a similar system ,concluded at providing an estimate to the tweet that is input to the system . The estimate that is provided as an end result is in terms of a probability of occurrence of the event "Earthquake" .The probability states that the chances of a tweet confirming the event. The difference of the probability from thresholds at both resulted in giving response as "EVENT DETECTED" or "EVENT NOT DETECTED ". The probability is based on the dataset of the system that is created using past results. The research stated in this paper can be further taken into consideration to act in the real time i.e., to send alert responses to the user based on their Geo location.

The proposed model can be used in real time with some modifications to help people act in case of earthquake. The project topic demands further research in future.

7. FUTURE SCOPE

In future, all the individual modules can be collected into a complete application. Thus it can be considered to be the stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The application can be developed using NetBeans as front end and SQL server is used as backend. We can create a user page using GUI, where in user can send message from one source to another. User can read the messages from many different sources and also user can maintain their personal information. He/she can identify the tweet having information about earthquake and will detect the target event.

a) **Authentication**: In Authentication module, if new user is going to login to consume the service then the user has to register first by providing necessary details. After successful completion of sign up process, the user has to login to the application by providing username and password.

b) **Profile Manager**: In this module, user can create a profile to maintain their personal information and also can customize the tweet subscription process. The personal information involves address, contact number, profile picture, occupation etc.

c) **Tweet Subscription**: In this module, registered user can subscribe to the tweets from different sources by following the genuine tweet maker. Once subscribed, all future updates are sent to the subscriber through electronic communication. There are four types of response actions performed in tweet subscription which are compose tweet, read tweet, following and profile setting.

d) **Alert sender**: This module helps to distribute the alert message to twitter recipients through electronic communication.

ACKNOWLEDGEMENT

The research on "Real Time Event Detection" has been given to us as a part of the curriculum in 4-Years Bachelors Degree in Computer Science.

We have tried our best to present the project as clearly as possible using basic terms that we hope can be comprehended for further studies.

We have completed this project under the able guidance and supervision of Prof. Isha Pathak. We will be failed in our duty if we do not acknowledge the esteemed scholarly advice, assistance and knowledge, we received from her towards fruitful and timely completion of this work.

We are also immensely grateful to our friends for sharing their pearls of wisdom with us during the course of this research project.

REFERENCES

[1] A. Java, "Why We Twitter : Understanding Microblogging Usage and Communities," Proc. Ninth WebKDD and First SNA-KDD Workshop Web Mining and Social Network Analysis (WebKDD/SNA-KDD '07), pp.

[2] "Social Networks that Matter: Twitter Under the Microscope," ArXiv E-Prints, http:// arxiv.org/abs/0812.1045.

[3] A Survey of Techniques for Event Detection in Twitter: http://onlinelibrary.wiley.com/doi/10.1111/coin.12017/abstract

[4] Earthquake Shakes Twitter Users: Real-Time Event Detection By Social Sensors. Proceedings of the 19th International conference on Word Wide Web, 851–860

[5] NaïveBayes: https://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/

[6] TwitterDocumentation. https://dev.twitter.com/rest/public

[7] TweepyDocumentation: http://docs.tweepy.org/en/v3.5.0/, https://github.com/tweepy/tweepy

[8] Ozdikis, O., Senkul, P., and Oguztuzun, H. (2012). Semantic expansion of hashtags for enhanced event detection in Twitter. In Proceedings of the 1st International Workshop on Online Social Systems.