SOCIAL MEDIA DATA ANALYSIS FOR TRAFFIC INCIDENT DETECTION AND MANAGEMENT

Kaiqun Fu
Department of Computer Science
Virginia Polytechnic Institute and State University
7054 Haycock Rd, Falls Church, VA 22043
Tel: 703-538-8310; Email: fukaiqun@vt.edu

Rakesh Nune, Corresponding Author
District of Columbia Department of Transportation
55 M Street, SE, Washington DC20003
Tel: 571-265-4047; Email: Rakesh.Nune@dc.gov

Jason X. Tao
District of Columbia Department of Transportation
55 M Street, SE, Washington DC20003
Tel: 202-671-1489; Email: jason.tao@dc.gov

Word count: 2,527 word text + 6 tables/figures x 250 words (each) = 4,027

July 31, 2014

ABSTRACT

Social media and online services with user-posted content have generated a staggering amount of information that has potential applications in various areas such as sentiment analysis and emergency management. In this research, we explored the feasibility of using social media data for detecting traffic incidents and collecting supplemental incident information. A comprehensive approach has been developed to extract and analyze real-time traffic related twitter data for incident management purpose. The developed approach consists of three steps: (1) Development of traffic incident related key words and their association rules; (2) Extraction of real-time tweets with influential word sets; and (3) ranking and selection of traffic related tweets. The developed approached was implemented at District of Columbia Department of Transportation for incident management. Data validation has been conducted against the real-world incident database. The preliminary results of analysis have shown that social media data is promising for early incident detection and can be used as supplemental source for incident data collection.

Keywords: Social media analysis, Incident Detection, Traffic Incident Management, Tweet extraction, Term frequency

INTRODUCTION

Early detection of traffic incidents is critical to reduce the impact of incidents on the traffic conditions. There are multiple incident detection sources which include reports of roadway operation patrollers, citizen calls, CCTV monitoring and alerts from vehicle detection stations. In the District of Columbia, the CCTV system is the most important means to detect and verify traffic incidents. Since the current CCTV system does not provide 100% coverage in the District, delay in incident detection may take place especially in those areas not covered by the CCTV system. Recently, we conducted a pilot project to explore the application of social media data analysis in incident detection and traffic management.

In the past decade, as social media (e.g., Twitter, Instagram and Facebook) became more and more popular, social media data has been collected and used in various applications. For instance, Bollen *et al*. (1) performed sentiment analysis on large scale Twitter data to predict the daily directions of stock market. Schmidt (2) explored the social media data to predict and track disease outbreaks. Sabra (3) explored the benefits of integrating social media tools into emergency communications and monitoring the social media contents during an emergency or disaster. Schulz *et al*. (4) developed an approach for efficient processing and storing of social media data for emergency management purpose. Applying social media analysis in traffic management is a relatively new concept. Today, many government organizations and news companies are using social media such as Twitter to communicate with the public and commuters on traffic incidents. On the other hand, many commuters report incidents, special events or congestion levels that on route through the social media. Therefore, the contents of popular social media contain abundant information on traffic incidents and roadway congestion. It is feasible to extract and analyze the social media data for detecting traffic incidents and obtaining supplemental incident information.

In this pilot project, we developed a comprehensive approach to extract and analyze real-time traffic related twitter data for incident management purpose. The developed approach consists of three steps: First, a set of keywords were extracted from historical traffic related tweets generated by influential users, then a comprehensive algorithm is applied to establish the association rules of keywords; Secondly, real-time tweets are extracted through Twitter API with influential word sets; At the last step, ranking and selection of the extracted tweets are conducted to capture the most relevant incident tweets. A detailed methodology is presented in the next section, followed by a description of implementation of the developed approach and data validation. Conclusions are drawn in the last section.

METHODOLOGY

In order to successfully extract traffic incident related tweets, it is necessary to establish a set of keywords that will be sent over twitter to query data through the twitter timeline API [5]. Through preliminary analysis, we have identified four influential Twitter users who actively post traffic information. These four account names are “WTOPtraffic”, “VaDOT”, “drgridlock” and “DCPoliceDep”, which are held by WTOP, Virginia DOT, The Washington Post and District of Columbia Police Department, respectively. The tweets from these four users are used to develop

the set of keywords. In this research, a data collector is built to collect 3200 recent tweets posted by four influential users that compose a document set, denoted as T_I.

A statistic method called "term frequency–inverse document frequency" (or simply as *tf-idf*) (6) is applied to document set T_I to determine the importance of each word in the set. *tf–idf* is the product of two statistics, term frequency and inverse document frequency. For word *t* and document d, the term frequency *tf*(*t*,*d*) is defined as:

$$tf(t,d) = \frac{f(t,d)}{\max\{f(w,d): w \in d\}} \quad (1)$$

where *f*(*t*, *d*) is the number of times that term *t* occurs in document *d.*

The inverse document frequency, or *idf*, is a measure of how much information the word provides, that is, whether the term is common or rare across all documents. *idf* is defined as

$$idf(t,D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right) \quad (2)$$

where D is a corpus of all document, N is the number of documents in the corpus, and $|\{d \in D: t \in d\}|$ is the number of documents where word t appears. In our case, each tweet is a document, and all tweets together compose a corpus. Since all words to be analyzed are from the extracted tweets, $|\{d \in D: t \in d\}|$ will never be zero. We choose a total of 50 words with highest *tf-idf* weights as the traffic incident related key words. These key words are shown in Table 1.

TABLE 1 List of Traffic Incident Related Keywords

#dctraffic	#mdtraffic	#vatraffic	crash	due
delays	st	traffic	right	ave
lane	lanes	rd	nw	accident
buses	left	md	nb	bridge
sb	earlier	street	near	blocked
loop	congestion	expect	va	dc
update	road	work	following	closed
open	vehicle	inner	car	killed
new	get	minute	directions	close
schedule	police	beltway	operating	us

Use Q to denote the set of all above keywords. The keywords list Q can be used as input to the Twitter Search API to crawl all the tweets that match the queries.

Through the preliminary analysis, it is found that if a single keyword such as “crash” or “lane” is used in a query, there are enormous noises in extracted tweets. In addition, the number of tweets from the data crawler will easily reach the assigned rate limit. In order to solve these issues, we applied the Apriori algorithm (7) to keyword list Q to develop the association rules of keywords and then construct Twitter queries using a combination of a few keywords (or called “wordset”) rather than single keyword.

The Apriori Algorithm is usually applied in the field of transaction mining to establish frequent itemsets for boolean association rules. The algorithm proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. In general, the concept of the Apriori Algorithm can be expressed in the following steps:

Step 1: Identity frequent itemsets that are the sets of item with minimum support (denoted by L_k for kth-Itemset);

Step 2: Apply the Apriori property: Any subset of frequent itemset must also be frequent. That means all subset of frequent itemset must have minimum support;

Step 3: Extend the length of itemset by the join operation: To find L_{k+1} , a set of candidate (k+1)-itemsets is generated by joining L_k with itself.

In this case, each keyword is viewed as an item and thus a wordset is viewed as an itemset in the field of transaction mining. The minimum support level is set to 0.02 in this implementation. It is also found that when wordsets of 3 keywords are used, the algorithm is able to filter most noise in tweets and the results are optimized.

In order to highlight the most relevant tweets from the extracted dataset, all tweets are ranked based on the tf–idf weights of the contained keywords. For each tweet, a score is calculated by adding the weights of all keywords appearing in the tweet. All extracted tweets are ranked by their scores and stored into the database.

IMPLEMENTATION AND ANALYSIS

The above described approach for Twitter data extraction and analysis was implemented at District of Columbia Department of Transportation in late 2013. An intuitive user interface is built to display the latest traffic tweets along with visualizations tools like histogram. The main front end user interface was implemented using bootstrap (8) - a HTML5 and CSS3 based web applications framework and backend with Django (9). The implemented user interface allows the users to search for the incident or congestion related tweets on particular dates and times. This feature helps the operators at traffic management center to obtain supplemental incident information which may be missing from the existing incident profiles. Tweets volume histograms for a regular work day and a weekend day are displayed in Figures 1 and 2. The figures clearly shows that in a regular work day, more traffic related tweets are posted during morning and afternoon rush hours. In a weekend day, most traffic related tweets are posted in the evening time from 5:00PM to

8:00PM. This is consistent with the travel rates and incident rates over daily time.

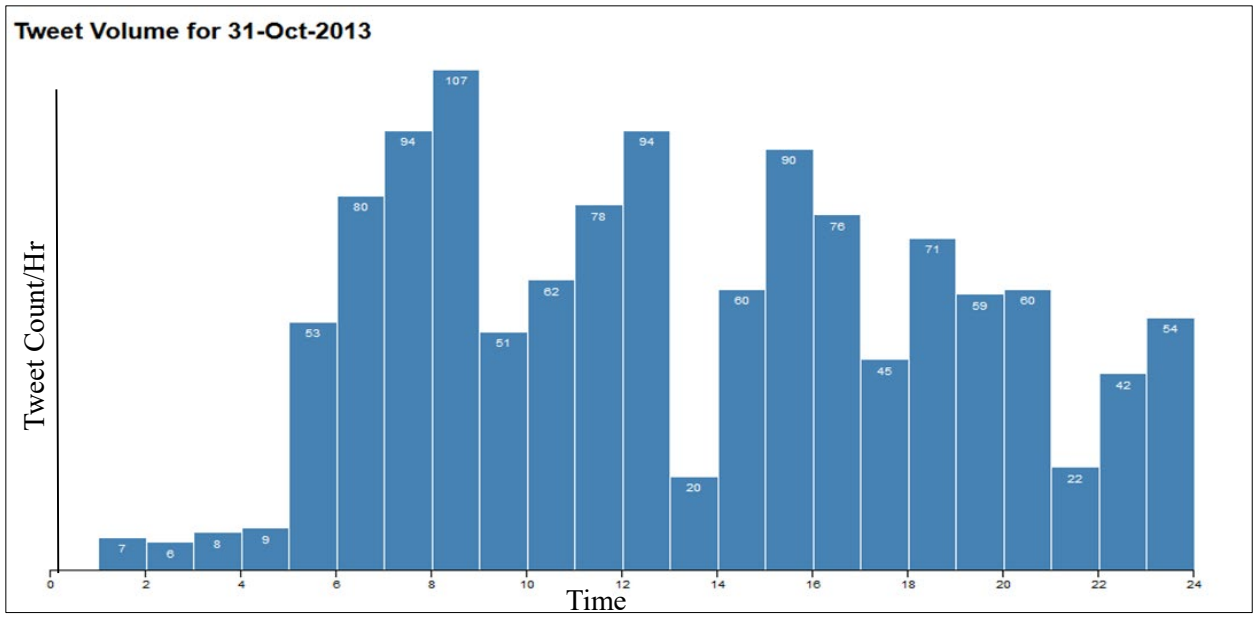


FIGURE 1 Tweets Volume Histogram on a Regular Work Day

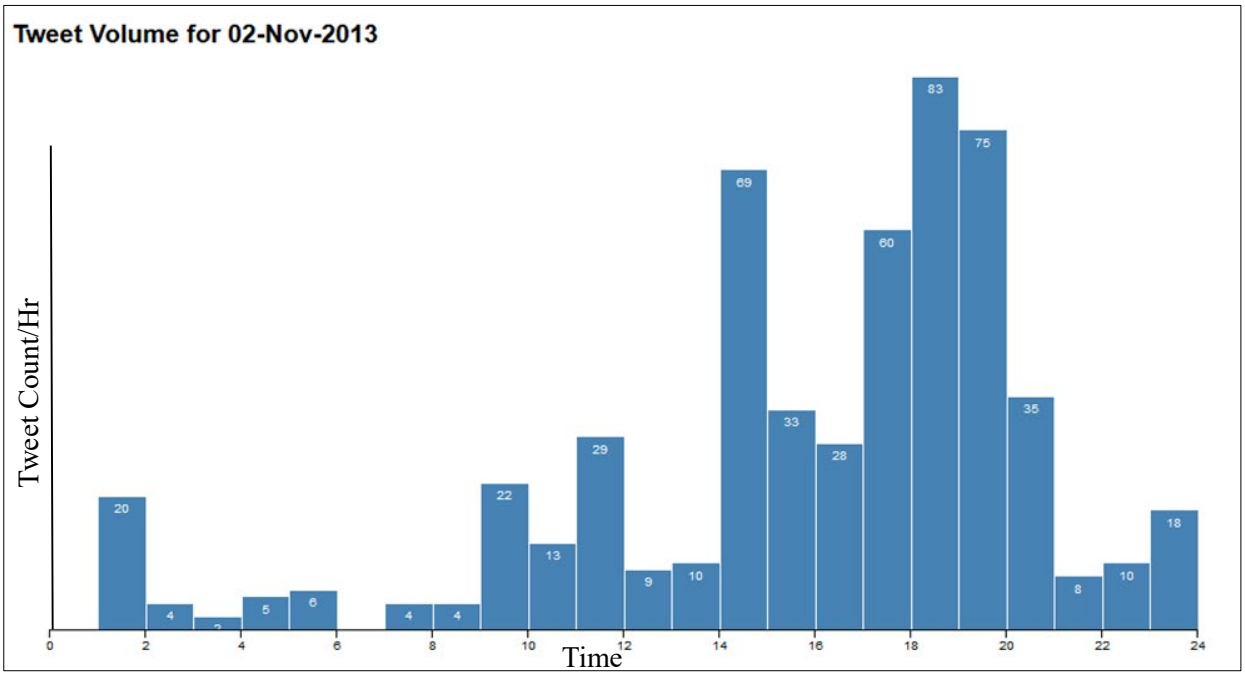
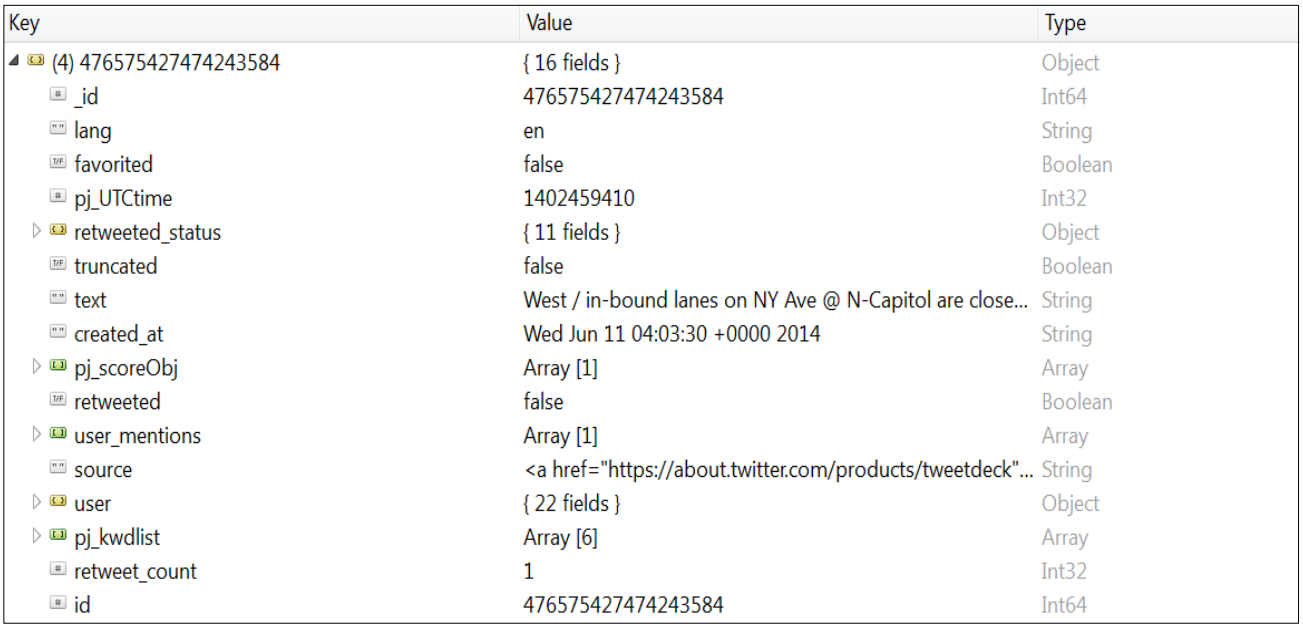


FIGURE 2 Tweets Volume Histogram on a Weekend Day

The twitter data is unstructured and number of fields varies depending on the tweets unlike a predefined model such as relational databases. The number of variables may changes depending

on users' profiles and input. To deal with the scalability issue and storage requirements, NoSQL database MongoDB [10] is used to store and process the extracted tweet. Figure 3 shows the database schema used in NoSQL.



Key	Value	Type
(4) 476575427474243584	{ 16 fields }	Object
_id	476575427474243584	Int64
lang	en	String
favorited	false	Boolean
pj_UTCtime	1402459410	Int32
retweeted_status	{ 11 fields }	Object
truncated	false	Boolean
text	West / in-bound lanes on NY Ave @ N-Capitol are close...	String
created_at	Wed Jun 11 04:03:30 +0000 2014	String
pj_scoreObj	Array [1]	Array
retweeted	false	Boolean
user_mentions	Array [1]	Array
source	<a href="https://about.twitter.com/products/tweetdeck"...	String
user	{ 22 fields }	Object
pj_kwdlist	Array [6]	Array
retweet_count	1	Int32
id	476575427474243584	Int64

FIGURE 3 The Database Schema in NoSQL

Validation has been conducted by comparing the real-time tweet contents extracted by the developed algorithm with the real-time incident data collected through other sources (e.g., CCTV monitoring, roadway patroller reports and citizen's call). The results of analysis have shown that the algorithm is able to capture almost all vehicle crashes or disabled vehicles on the roadways that cause congestion. One of the important findings is that the Twitter data contains much more numbers of incidents than that collected through the CapTOP system – a traffic management platform used by the District Department of Transportation.

In order to illustrate the efficiency of the developed approach, we compared some of the detected incidents that took place on June 8, 2014 and June 9, 2014 between the CapTOP database and the extracted tweets. Table 2 lists four incident records in the CapTOP database that were reported by roadway operation patrollers. These records include incident ID, detection date and time, incident location information (Street names) and incident descriptions and operators comments for these incidents. It can be seen that for most of the above records, the incident description and operator comments do not provide sufficient information for these incidents.

TABLE 2 Incident Records in the DDOT Incident Database

Incident Id	Date	Time	Street Id (A)	Street Id (B)	Description	Comment
69673	6/18/2014	09:08	Branch Ave	Pennsylvania Ave	DC eg 3526	NULL
69724	6/19/2014	10:50	First ST	Florida Ave	NULL	Nothing found ROP 4 responded to incident.
69734	6/19/2014	16:59	Park Street and W Street	Branch Ave	NULL	NULL
69735	6/19/2014	17:10	Eastern Ave	I-295	NULL	Slow traffic condition nb 295 prior to crash in Maryland side. Camera out of order

Figures 4(a) and 4(b) show the corresponding tweets for each incident listed in Table 2. From the Figure 4(a) and 4(b) one can see that the posted time for each tweet is significantly earlier than that recorded in the CapTOP database. More importantly, from these tweets users can capture much more detailed information about the incidents. Such supplemented information is not only traditionally textual, but also pictorial. For example, one of the tweet results for incident number 69734, posted by DC Police Department, contains a detailed screen shot of Google Map with the incident location highlighted. Such visual and user friendly information presentation can enhance the DDOT incident database with the concept of *Big Data*.

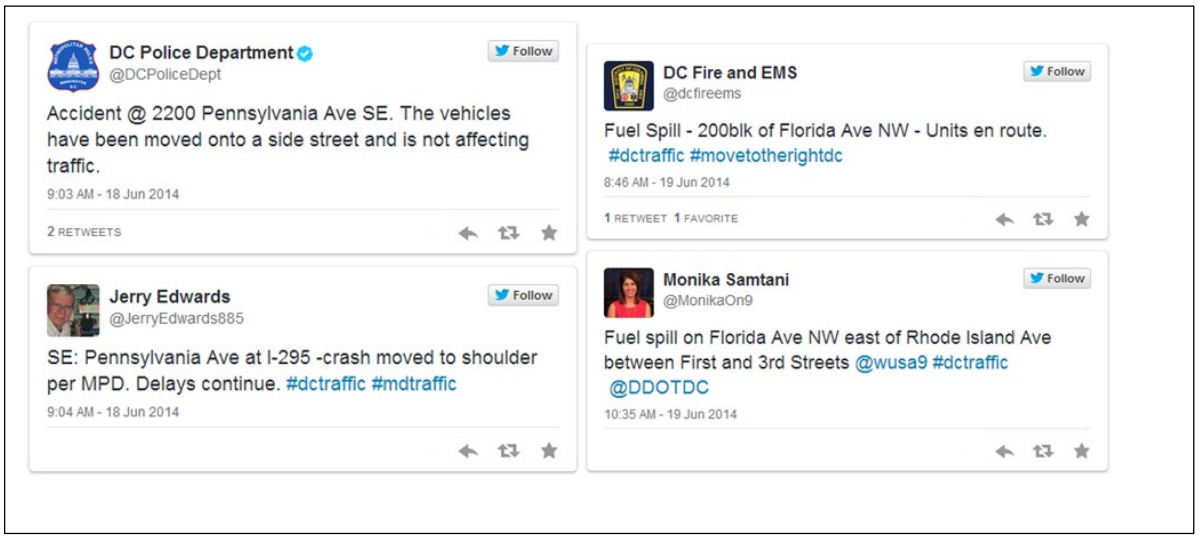


FIGURE 1a Traffic Incident Related Tweets Extracted from the Algorithm

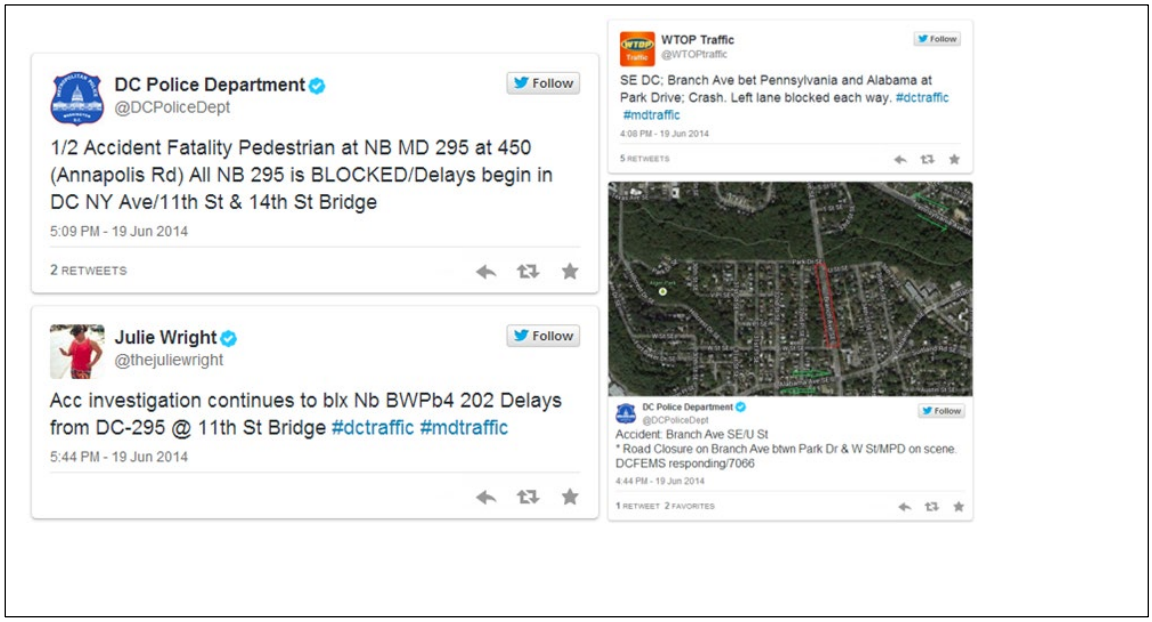


FIGURE 2b Traffic Incident Related Tweets Extracted from the Algorithm

CONCLUSIONS AND FUTURE WORK

In this research, a comprehensive approach has been develop to extract and analyze the traffic incident related tweets for detecting traffic incidents and collecting supplemental incident information. The validation and analysis has showed the promising application of the approach in incident management.

The twitter data crawling and scoring process proposed in our algorithm utilizes *tf-idf* calculation to determine the weights for each keyword. The algorithm used currently is static, which means once the weight for each keyword is calculated; the weight will be assigned to the keyword for all following tweet score calculations. Although this method can provide better algorithm performance, it cannot model the actual transportation events. Since the appearance of transportation events are random, and more importantly independent to each other. In other words, knowledge from the previous events gives no inference to the next. Due to this characteristic of the transportation events, it is optimal to update the keywords weights dynamically. This scheme is planned to be done in our future works, a relatively short time section will be defined and the weights for keywords will be calculated and updated in this time section. Then the updated weights will be used in the tweet score calculation for the next time section.

All the code developed for above analysis is available for download from DDOT-TOA Github page (http://ddot-toa.github.io/).

REFERENCES

1. Bollen, J., H. Mao, and X. Zeng. Twitter mood predicts the stock market. In *Journal of Computational Science*, Vol. 2(1), 2011, pp. 1-8.
2. Schmidt, C. W., Trending Now: Using Social Media to Predict and Track Disease Outbreaks. In *Environmental Health Perspectives*; 2012, Vol. 120 Issue 1, pp30-33.
3. Sabra, J. Emergency Management 2.0: Integrating social media in emergency communications. In *Journal of Emergency Management*, Vol. 9(4), July/August, 2011, pp. 15- 18.
4. Schulz, A., J. Ortmann, F. Probst, Getting user-generated content structured: Overcoming information overload in emergency management. In *Proceedings of 2012 IEEE Global Humanitarian Technology Conference (GHTC 2012),* 2012, pp.1-10.
5. Makice K., *Twitter API: Up and Running: Learn How to Build Applications with the Twitter API*. O'Reilly Media, 2009, ISBN-10: 0596154615.
6. Rajaraman, A., and J. D. Ullman (2011). *Mining of Massive Datasets*. Cambridge University Press, 2011, pp. 1–17.
7. Agrawal, R. and R. Srikant, Fast Algorithms for Mining Association Rules in Large Databases, in *Proceedings of the 20th International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., 1994, pp. 487-499.
8. Spurlock, J. *Bootstrap*. O'Reilly Media, 2013, ISBN-13: 978-1449343910
9. Alchin M. *Pro Django*, Apress, 2013, ISBN-13: 978-1430258094
10. Hoberman, S. *Data Modeling for MongoDB Paperback* (*First edition*). Technics Publications, 2014, ISBN-13: 978-1935504702