Accepted Manuscript

Towards real-time event detection for online behavioral analysis on social big data

Duc T. Nguyen, Jai E. Jung

PII: S0167-739X(16)30089-9
DOI: http://dx.doi.org/10.1016/j.future.2016.04.012
Reference: FUTURE 3012

To appear in: *Future Generation Computer Systems*

Received date: 13 March 2016
Revised date: 14 April 2016
Accepted date: 20 April 2016

Please cite this article as: D.T. Nguyen, J.E. Jung, Towards real-time event detection for online behavioral analysis on social big data, *Future Generation Computer Systems* (2016), http://dx.doi.org/10.1016/j.future.2016.04.012

Towards Real-time Event Detection for Online Behavioral Analysis on Social Big Data

Duc T. Nguyen[a], Jai E. Jung[b,*]

[a] *Faculty of Information Technology, Vietnam Maritime University, Hai-Phong, Vietnam*
[b] *Department of Computer Engineering, Chung-Ang University, Seoul, Korea*

Abstract

Social Networking Services are increasingly becoming popular for Internet citizens in their daily life, especially since the advent of smart mobile devices which are integrated with utility modules such as 4G/WIFI connectivity, Global positioning services, Cameras, Heart beat sensors, and so on. It is easy to come across the use of such devices for sharing information at anytime which can be listed as posting photo, sharing a status, or narrating an event. The behavior of users makes the flow of data (or a Social Data Stream) has real-time characteristics which actually are notifications about your friends's posts after a short delay of diffusion over the network. Inside the data stream, news pieces related to real social facts are covered together with unfocused information. And outstanding facts (or events) surely draw more public attentions, it is evidenced by the number of relevant messages or communication interactions between interested persons toward certain topics. Technically, the characteristics of data in the aforementioned scenario gives us an opportunity to build a model which can automatically determine occurrences of events from the Social Data Stream. In this paper we propose an approach to solve the challenge of early event identification, which requires proper approaches to process incoming data in terms of processing performance and number of data.

Keywords: Social Network Analysis; Event Detection; Real-time event

*Corresponding author
Email address: j3ung@cau.ac.kr (Jai E. Jung)

detection.

1. Introduction

We are now living in a thoroughly connected world where various kinds of information are generated and shared daily over the Internet. Several web-based platforms allow people to interact and share their interests and activities through short messages, including video, photo and textual information. Social Networking Services (also Social Networking Sites or SNS) are such platforms. Nowadays, these have already become the most popular environment for communication. Whenever a new message is generated and shared on SNS, users unintentionally describe life around them under their own perspective. It is therefore very interesting if the number of discussions and shares of a certain content are suddenly increased in a short time. The such unusual situations are a form on instruction to predict occurrences of a particular trend of users, who focus on a certain kind of news content and contribute to propagate the information to their friends by using functions of SNS such as "Shared", "Reply", "Like" buttons in Facbook or "Retweet", "Mention" functions in Twitter.

In recent years, there are more than 288 million monthly active users on Twitter who post more than 500 million tweets per day, 1.35 billion monthly active users are on Facebook, and 187 million active monthly users are on LinkedIn and so on. Aiming to inform the latest relevant messages to users or the third party clients, SNS introduce special service called data stream, it automatically sends a notification message to related users of relevant messages if any. Therefore, the data has temporal characteristic and its content is evolved from time to time. Obviously, it is useful if we can automatically extracting events in real-time by listening on the input resource. However, it also opens technical issues such as how to improve the processing performance and efficiency of the event detection method on the dynamic data resource.

However, traditional topic detection approaches are not designed to detect the kind of events efficiently in real-time, particularly if the data sources are

influenced by noise data and the content contains diverse topics. To overcome the issue, the paper proposes a model for extracting and tracking events from a given Social Data Stream in real-time, which combines content-based features of original textual data and the propagation of news between viewers into discrete signals. We analyze abnormalities in the signal oscillation over time and the number signals of the same kind to identify the approximate time period of events.

2. Related Work

For detecting new topic occurrences, previous researches usually try to find the abnormality in the collected a data. The abnormalities is various ranging from the irregular density of relevant documents, the repetitive usage of certain keywords, to the changes in daily routines of data and so on. Generally, the collected data is broken into smallest pieces called *terms* or *keywords* which is material for clustering the data using two approaches: ***i) Content-based*** [1, 2, 3, 4, 5], ***ii) Feature-based*** [6, 7, 8, 9]. In [10], the authors indicate that new event detection refers to identify the first story on a given topic of interest by constantly monitoring news streams. Aiello et al. compared six topic detection methods using three Twitter datasets that had been collected using the Twitter API for three major events including the 2012 FA Cup final, the "Super Tuesday" primaries in the presidential nomination race for the US Republican Party, and the US Election in November 2012 [11]. They recognize that a method based on n-grams co-occurrence provided better results for the topic recall, keyword precision, and keyword recall.

Several research groups have attempted to use discrete signals to preprocess data in order to identify keywords in an SNS data stream. Only keywords, which have a high frequency or "burst" characteristic, are kept for event detection [6, 7, 8]. A clustering process is performed on the keyword signals to classify them into the corresponding categories to determine the occurrences of events. He et al. use a Discrete Fourier Transformation (DFT) method [12] to find peaks in the

frequency domain of the keyword signals after these were grouped in five feature types according to the power spectrum strength length and the periodicity [13]. Weng and Lee extend the method by using a wavelet transformation [14], the authors conclude that the temporal information of the signals is lost if DFT is used even though it is a significant property. Events are extracted from data sources by clustering keywords using modularity-based graph partitioning on the wavelet transformation. Both approaches analyze the evidence of events according to the strange volatility of the frequency of the keywords, but they have not considered the combination of these keywords in tweets to express the event features. In addition, users' reactions toward the news provided on SNS have not been examined.

Li et al. introduce an approach [9] to construct a system for extracting events by analyzing tweets, which are tracked from Twitter by using a certain kind of filter. Authors focus to identify Crime and Disaster related events by classify tweets with Social features such as Twitter-Specific features and the topic's features, but only content-based features are considered. Other meta data related to the diffusion of tweets are not adequately analyzed, meanwhile that data can express the level of interesting, solidarity of viewers toward a given topic at a certain time. In [15] and [9], the authors also use geographical location embedded in the tweets to enhance the event detection result. However, because of the privacy issues, not so much SNS users enabled the GPS feature on their smart devices. Less than 1% of tweets are geo-tagged and sometimes location information in SNS users profiles is unreliable due to the travel [16]. In addition events can be observed from multiple places via a broadcast program like TV Channel, Live stream, Website and so on. Thus the geographical location information embedded in tweets is not a critical factor in determining the events which has the sphere of influence on the wide geographical range, but it is good if the events happen and impact on a community of people in a certain locality.

3. Data Presentation

In this section, we discuss how to transform a textual dataset into dataset of discrete signals. Given a temporal corpus of news or messages that are generated by users on an SNS, the messages can be distributed in a range of time T=[t1:t2]. We count the number of messages with a ΔT fixed rate that is the interval gap between two adjacent sampling positions. The ΔT unit decides the length of the signals that are generated for each feature in the period T that is considered. If using a ΔT small interval, the system need more memory to express signals, and the amount of processing time is increased relatively. On the positive side, it can generate an accuracy detection result in considering to the latency for system can identify events since the moment of its occurrences by analyzing abnormalities in the signals' oscillation.

3.1. Data Presentation

Definition 1 (Micro-text)**.** *A twt micro-text is a short textual message posted on a Social Network Service. Given that U^* is the closure set of the SNS users, the formula for micro-texts is*

$$twt \equiv (a, U_a, V, t) \tag{1}$$

where $a \in U^*$ is the author of the micro-text, $U_a \subseteq U^*$is a set of the author's friends, t is its time of the posting and V is a set of terms included in the content.

Definition 2 (Social Reaction)**.** *A social reaction to a twt micro-text is an action performed by an user in terms of diffusing the original message to the other users on his/her connection network. The reactions to the twt are represented as a set of R^{twt} micro-texts*

$$R^{twt} \equiv \{twt_i\} \tag{2}$$

where each twt_i is a micro-text related to the original twt. The solidarity relation is managed transparently from the users, depending on the characteristics of each SNS.

Definition 3 (Discrete Social Data Stream)**.** *A DS discrete social data stream is a discrete representation of a Social Data Stream. A continuous temporal sequence of micro-texts collected from a specific time t_0 is sequentially broken into batches of micro-texts based on the time they were posted. It means that the data stream is discretized by sampling the points separated by an ΔT interval. DS is an ordered sequence, its formula is*

$$DS \equiv (B_1, B_2, ..., B_i, ...) \tag{3}$$

where each batch $B_i = \{twt_j\}$ is a set of micro-texts collected in the sampling slot $i-th$ from $t_0 + i \times \Delta T$ until $t_0 + (i+1) \times \Delta T$ and $B_j = \emptyset$ in case $j < 0$.

From time to time, the size of the collected data increases to such an extent that it cannot be handled and processed immediately. This is an unexpected situation of our real-time event detection method. In addition, a news topic exists only in the short term, so we don't need to analyze all the data to identify events but might consider a wide enough range of time. For such a purpose, we introduce a "Sliding Window" concept to shift forward by a given amount according to the time interval ΔT. Only data collected in the sliding windows are processed to extract new information.

Definition 4 (Sliding Window)**.** *A Sliding Windows is a window of time T_k refers to a specific period where the data collected from the stream can be instantly monitored and processed.*

Given a window of time T_k, the data collection of the sliding window includes all messages collected in the given period on a DS discrete social data stream. For this ordered sequence of micro-text sets, the formula is defined as

$$DS\left[T_k\right] \equiv \left(B_{k-N+1}, B_{k-N}, ..., B_k\right). \tag{4}$$

In order to determine the significance of each keyword in a given dataset, a measurement based on their occurrence is often calculated. The Term Frequency - Inverse Document Frequency (tf.idf) method is a numerical statistic that is

intended to score the importance of a keyword in a document [17]. However, this method is not suitable to calculate the information in a dataset collected from an SNS. The SNS messages are often short, and each keyword might occur only one time. Therefore, we prefer to consider a group of messages as one document when using this approach. The groups are defined in terms of the data that is collected for each sample. The samples have temporal characteristic, so the documents are ordered sequences in terms of time. Hence, the score for a given keyword that is computed on the sequence is a variation of its importance indexed by the sample positions. It forms a discrete signal where each peak is a potential candidate for the occurrence of an abnormal thing that is described by the keyword. In addition to measuring the occurrence feature of each of the keywords, we propose to consider user reactions toward messages containing a keyword due to it could give a significance level of the keyword to attract the attention of viewers. The diffusion speed of original messages between users is required as well for the analysis. In another words, it is more important to consider the probability that several keywords are main factors to express a certain fact so that users use them frequently and be strongly attracted by messages contained it.

Definition 5 (Keyword's Score)**.** *Given a corpus $DS[T_k]$ of micro-texts collected from a certain SNS, V^* is the closure of terms extracted from $DS[T_k]$. The score of a term $w \in V^*$ at an $i-th$ sample is a real value in range $[0:1]$, it is defined with a multivariate function as following*

$$S_w(k,i) = \mathcal{G}\left(S_{w,c_1}(k,i), S_{w,c_2}(k,i), ..., S_{w,c_j}(k,i), ...\right) \ \forall c_j \in C \tag{5}$$

where C is a set of characteristic features related to the diffusion of messages on the SNS, S_{w,c_j} is the particular score of the term w that is considered under the context feature c_j.

In this paper, we only consider three major features of the information which are diffused, but a general case is not limited to such. The features include the occurrence, which is the degree to which keywords appear over a given time;

diffusion-degree, which is the affecting level of diffused news calculated according to the number of participants; diffusion-sensitivity, which is the speed at which the information spreads from a user to the followers.

Definition 6 (Occurrence Score)**.** *Given a corpus of micro-texts $DS[T_k]$, the occurrence score of a certain term w is calculated using an extension of tf.idf as follows*

$$S_{w,Occcurence}(k,i) = \frac{|B_i^w|}{|B_i|+1} \times log \frac{\left|\bigcup_{B_j \subseteq DS[T_k]} B_j\right|}{\left|\bigcup_{B_j^w \subseteq DS[T_k]} B_j^w\right|+1} \tag{6}$$

where B_i^w, B^w are the subset of micro-texts that contain the given term w, and all micro-texts that are collected at the sample indexed i in the corpus, respectively.

Definition 7 (Diffusion Degree)**.** *Given a corpus of micro-texts $DS[T_k]$, which is the diffusion degree of a term w at a sample, is the probability of being of concern to SNS users during the given time period. It is calculated as*

$$S_{w,D.Degree}(k,i) = \frac{|\bigcup_{twt^w \in B_i} R_i^{twt^w}|}{\left|\bigcup_{twt \in DS[T_k]} R^{twt}\right|+1} \tag{7}$$

where $R_i^{twt^w}$ is a set of user reactions that are collected at the sample indexed i toward a micro-text twt^w that contains the term w, and R^{twt} is a set of user reactions to a micro-text in $DS[T_k]$.

Definition 8 (Diffusion Sensitivity)**.** *Given a corpus of micro-texts $DS[T_k]$ and B_i is the subset of micro-texts that are collected at the sample indexed i. The diffusion sensitivity score of a term w at the sample is the probability of how fast the information diffuses from the author to other users during the given time period. It is calculated as*

$$S_{w,D.Sensitivity}(k,i) = 1 - \frac{\underset{\forall twt^w \in B_k, \forall twt \in R_k^{twt^w}}{AVG} t_{twt} - t_{twt^w}}{\underset{\forall twt^w \in DS[T_i], \forall twt \in R^{twt^w}}{MAX} t_{twt} - t_{twt^w}}. \tag{8}$$

3.2. Time Sequence Presentation

Definition 9 provides the formula to represent the signal form of a keyword generated using its score at each sample position, the scores are computed using

Definition 5. The strange areas in signals are significant evidence of abnormal occurrences.

Definition 9 (Keyword's Signal)**.** *The signal of an individual term w is represented as a discrete sequence of its meaningful degree at a sampling position. Its formula is*

$$w^k(i) \equiv S_w(k,i), \qquad \forall i \geq 0 \tag{9}$$

where $S_w(k,i)$ is the score of the term w at the sample indexed i in a given corpus $DS[T_k]$.

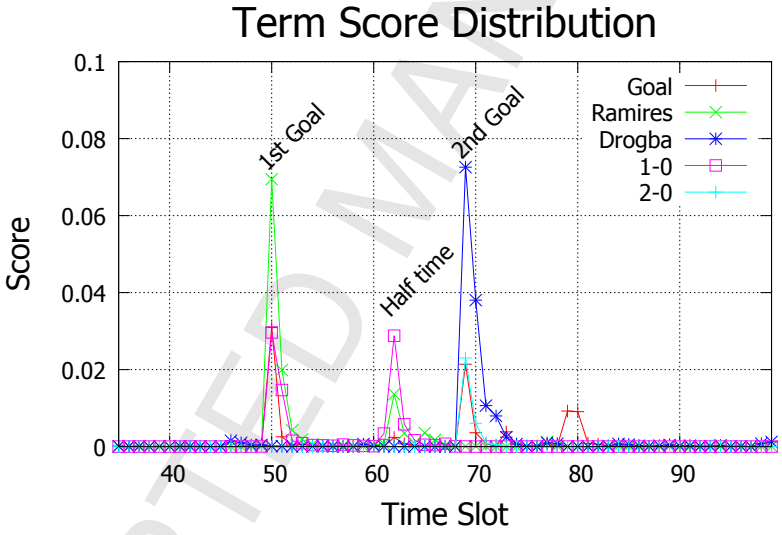


Figure 1: Distribution of Terms' scores

Figure 1 is an illustration of the concentration area of the signals' power where high peaks are the most likely candidates for the occurrence of the events. It also shows that the relevant keywords often overlap at these areas. Definition 10 provides a way to combine signals of the micro-text keyword into a single one. The combined signal improves the power of its component in the areas that are most overlapped in order to determine the moments in the irregular area where the keywords' signals oscillate with a similar pattern.

Definition 10 (Micro-text's Signal)**.** *The temporal sequence form (or signal) of a micro-text twt is defined through a $\mathcal{F}$ semantic combination function based on the signal of its terms as follows*

$$twt^k(i) \equiv \mathcal{F}\left(w_j^k(i)\right), \qquad \forall w_j \in V_{twt} \tag{10}$$

where $S_w(k,i)$ is the score of the term w at the sample indexed i in a corpus $DS[T_k]$.

In general, we assume that the number of N samples taken on the given corpus $DS[T_k]$ is an exponential value of 2 by manipulating the sampling rate. This feature allows the Fast Fourier Transformation (FFT) to be applied on the temporal sequence of terms or micro-texts [12, 18], and we denote the relevant concepts as follows.

- $W(x), TWT(x)$ are the Fast Fourier Transformations of $w(i)$ and $twt(i)$ discrete sequences, respectively.
- The signal power of a $w(i)$ signal with length N is given as

$$P_w = \frac{1}{N} \times \sum_{\forall x \in [0:N]} |W(x)|^2 . \tag{11}$$

- The power spectral density or the distribution of the signal power on each frequency of a $w(i)$ signal with length N is computed as

$$PSD_w(x) = \frac{|W(x)|^2}{N}. \tag{12}$$

3.3. Event Formalization

In Section 1, we discussed the definitions of an event from several perspectives. The presence of an event is the manifested in the number of participants who join to discuss, the speed with which users respond to the original news, the consensus in terms of the users' emotions, and so on.

Definition 11 (**Event**)**.** *A social event on an SNS is an abnormal phenomenon that exhibits characteristic features in its data stream. It corresponds to a topic*

that quickly draws attention from SNS users, as evidenced by the number of users reactions and the diffusion capability of the topic in a certain time period. Assume that e is an event that occurs during a time period $\mathcal{T}_e = [t_1 : t_2]$ *(or also the sampling period* $\Gamma_e = [i_1 : i_2]$*) and* Ψ_e *is a set of relevant micro-texts, the formula of event e is represented as*

$$e \equiv (\Psi_e, \mathcal{T}_e, \theta_e) \tag{13}$$

where θ_e *is a distribution of micro-texts in* Ψ_e *over the sampling period* Γ_e*. Denoting* $\theta_e(i)$ *is a set of micro-texts collected at the sample indexed i.*

According to Definition 11, we construct two principles based on the normal distribution to determine when an event occurs and what time it vanishes as follows.

- **Presence Condition**: If the number of related members admitted at a certain sample position i_1 overcomes a threshold compared to the previous samples of its members distribution sequence.

$$|\theta(i_1)| \geq \mu_\theta - \beta \times \sigma_\theta \quad with \quad \beta > 1. \tag{14}$$

- **Decay Condition**: If number of admitted members decreases under a threshold when compared to the normal admission rate.

$$|\theta(i_2)| < \mu_\theta - \beta \times \sigma_\theta \quad with \quad \beta > 1. \tag{15}$$

The coefficient β in equations 14, 15 is used to control the threshold. It is chosen as a value that is greater than 1 for determining the switching value of the distribution between the outliers and the normal values which are expected in the series. For convenience we denote $\mathcal{T}_e = [t_{occurrence}, t_{decay}]$ where $t_{occurrence}$ indicates the timestamp to start track a candidate cluster as an event, and t_{decay} corresponds to the decay timestamp of the event. The extracted events are ranked by examining its member admission rate.

4. Event Detection

4.1. Workflow

Basic tasks of our proposed method for detecting events from a Social Data Stream are described as following.

- **Collection Phase**: In practice, we manually limit the scope of the tweets collected from Twitter by using a search condition. Then, the data are cleaned to a normalized form, and all unnecessary elements are removed before storing it into the database using JSON format [19].
- **Extracting terms**: Normalized tweets are analyzed to extract all potential terms. We can improve this step by using Named Entity Recognition with predefined knowledge as the training data.
- **Building signals**: Term occurrence, its diffusion information are tracked accumulatively for each sample, as shown in Section 3.1. The statistical data obtained in the previous step is used to construct signals (temporal sequences of real values).
- **Extracting features**: This phase produces a semantic network between the tweets. It is a weighted directed graph where the nodes are tweets that include meta information and the edges between tweets are measured by the complement of the similarity degree that is calculated by applying the normalized cross-correlation function with a time lag of zero on their power spectral density. The direction of the edge is from a node that has a smaller value in terms of the timestamp.
- **Clustering**: For this phase, we apply density-based spatial clustering on a semantic network of tweets that is obtained in order to determine the potential clusters. The clusters include adjacent points which are close in terms of time and frequency. Each cluster is a potential candidate depending on the existence condition as described in Definition 11.

4.2. Detecting Events by Clustering

We are free to define a combination function $\mathcal{F}$ according to Definition 10. A combination function based on the semantic analysis of the collected tweets is a good suggestion. We define $\mathcal{F}$ is the minimization function to combine the signals for each time slot.

The major mission of the rest of this section is to cluster similar tweets into clusters. To identify these clusters, we introduce two concepts: i) The semantic distance expresses the closeness of two tweets with respect to time and similarity in the component frequencies, and ii) the semantic network presents the entire closeness relationship between the tweets.

Definition 12 (**Semantic Network Of Tweets**)**.** *A Semantic Network of Tweets is a weighted directed graph* $G =< \mathcal{V}, \mathcal{A} >$*that represent the solidarity between tweets in terms of the occurrence in time and the similarity of the component frequencies.*

- $\mathcal{V}$ *is a set of nodes that correspond to tweets associated with the meta-data (posting time, diffusion information and so on).*
- $\mathcal{A}$ *is a set of ordered pairs of nodes called edges denoted* (twt_i, twt_j)*. Each edge is a directed link from* twt_i *to* twt_j *and its weight expresses the closeness of nodes. It is computed as the complement of result of the cross-correlation function norm_corr applied to the power spectral density of the corresponding tweets.*
- *An edge* (twt_1, twt_2) *is existed if only if it satisfies the following condition:* $norm\_corr(twt_1, twt_2) \geq \gamma$ *and* $t_{twt_1} < t_{twt_2}$*.*
- *The neighbors of a node p are destination nodes of the edges starting from p.*

Definition 13 (**Semantic Distance**)**.** *Given a Semantic network of tweets* $G =< \mathcal{V}, \mathcal{A} >$*, the semantic distance between two nodes* twt_1*,* twt_2 *is a relative gap between them in a space of time and a similarity degree of component*

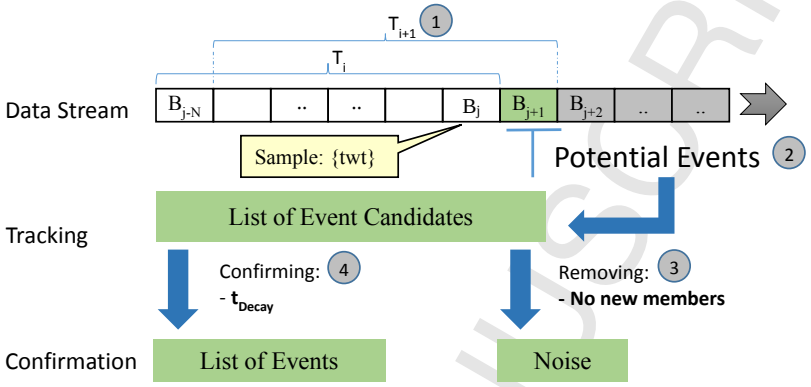


Figure 2: Workflow for detecting and tracking events

frequencies. The distance is computed as

$$d(twt_1, twt_2) = \begin{cases} \textit{Undefined} & \textit{if } \nexists(twt_1, twt_2) \\ \sqrt{[1 - sim_{twt_1,twt_2}]^2 + \delta t^2_{twt_1,twt_2}} & \textit{Otherwise} \end{cases} \quad (16)$$

where $sim_{twt_1,twt_2} = norm\_corr(twt_1, twt_2)$, *and* $\delta t_{twt_1,twt_2} = |t_{twt_2} - t_{twt_1}|$.

The semantic distance function $d(twt1, twt2)$ is the basic factor that is used to apply the OPTICS algorithm [20] to find clusters of tweets. Algorithm 1 shows the pseudo code to detect events from a given dataset Ψ.

4.3. Tracking Events

To track events in real-time, we need a suitable algorithm that works with the data stream. We consider each data collection $DS[T_i]$ of a certain time window T_i as a separate corpus. Algorithm 1 is applied on the $DS[T_i]$ to determine the candidates for events in the new approach. Algorithm 3 decides whether to continue to track the event or move it into an achievement list.

The workflow for Algorithm 3 is illustrated in Figure 2. It is an infinite loop that persistently remain open to a given data stream. During the process, tweets are collected into a temporal buffer, and after each fixed ΔT interval, the

Algorithm 1: DetectingEvent(Ψ, β, MinMembers, γ, ϵ, MinPts)

Input:

Ψ - a considered corpus

β - a threshold for determining events' time range

$MinMembers$ - a minimum number of tweets for forming an events

γ - a minimum threshold for forming edge between tweets

ϵ - a minimum distance for finding neighbors

$MinPts$ - a minimum number of neighbours

$\mathcal{V} = \emptyset$; $\mathcal{A} = \emptyset$; $G =< \mathcal{V}, \mathcal{A} >$

for *each pair* (twt_1, twt_2) *with* $twt_1, twt_2 \in \Psi$ **do**

  if $norm\_corr(twt_1, twt_2) \geq \gamma \wedge t_{twt_1} < t_{twt_2}$ **then**

    Appending twt_1, twt_2 into $\mathcal{V}$

    Appending (twt_1, twt_2) into $\mathcal{A}$ with the weight

    $norm\_corr(twt_1, twt_2)$

  end

end

$Candidates = OPTICS(G, \epsilon, MinPts)$

$E = DetermineEvents(Candidates, \beta, MinMembers)$

for *each* $e \in E$ **do**

  $M = \forall twt \in c \wedge t_{twt} \in [t_{occurrence} : t_{decay}]$

  $rank_e = \frac{|M|}{t_{decay} - t_{occurrence}(seconds)}$

end

Return the ranked list of events E

Algorithm 2: DetermineEvents(Clusters, β, MinMembers)

Input:

$Clusters$ - a list of cluster of tweets

β - a threshold for determining events' time range

$MinMembers$ - a minimum number of tweets for forming an events

---

$E = \emptyset$

for *each* $c \in Clusters$ *with* $|c| \geq MinMembers$ **do**

  Building the histogram h of member occurrences of c

  $t_{occurrence} = Undefined, t_{decay} = Undefined$

  Searching $t_{occurrence}$ and t_{decay} on h using equations 14, 15

  if $t_{occurrence} \neq Undefined \wedge t_{decay} \neq Undefined$ **then**

    $M = \forall twt \in c \wedge t_{twt} \in [t_{occurrence} : t_{decay}]$

    if $|M| \geq MinMembers$ **then**

      $E = E \cup \{(c, [t_{occurrence} : t_{decay}], h)\}$

    end

  end

end

Return the set of events E

buffer is stored into a queue as a block. The queue can only contain a maximum of N elements.

5. Experiments

In this section, we show an experimental result of the proposed model that detects events on Social Data Streams in real-time. It is necessary to examine the performance of the system tasks, which be improved by adjusting setting parameters. The latency of result is evaluated and compared against the real-timestamps of the events in order to show the applicability of the model in practice. The three major tasks of this section are therefore described as follows.

- **Reducing Noise Data**: Due to the nature of the data exchange over networks and in Social Networking Services that often take the form of short sentences and include many acronyms, all datasets collected from Social Media need to be preprocessed to clean out unnecessary data.
- **Evaluating the performance**: Section 5.2 evaluates the performance of the system. The system shows the capability to increase the processing speed to generate essential materials, such as signals of the keywords and tweets.
- **Evaluating the precision of the results**: Finally, the list of events that are extracted should be compared to the result obtained with another approach. In general, it is measured by the factor of the correct result with an expected list.

5.1. Preprocessing

Data from a social network usually contains a large amount of noise, including words that are misspelled, funny words, emotion-icons, hash-tags, URLs, and so on. The problem can be seen quite frequent in short messages.

Algorithm 3: RealtimeDetectingEvent(DS, T, ρ, β, MinMembers, γ, ϵ, MinPts).

Input:

DS - a considered data stream

T - a size for time windows

ρ - a threshold for enrolling new member

β - a threshold for determining events' time range

$MinMembers$ - a minimum number of tweets for forming an events

γ - a minimum threshold for forming edge between tweets

ϵ - a minimum distance for finding neighbors

$MinPts$ - a minimum number of neighbors group

$Finish = \emptyset, Tracking = \emptyset, i = 0$

while $DS \neq Empty$ **do**

  Collecting data $DS[T_i]$ corresponding with the time window T_i from DS

  Extracting all potential terms from $DS[T_i]$ into $Terms_i$

  Updating statistic information for each term in $Terms_i$ accumulatively

  Updating signals for each term in $Terms_i$

  Building signals for each tweet in $DS[T_i]$

  $C = DetectingEvents(DS[T_i], \beta, MinMembers, \gamma, \epsilon, MinPts)$

  $Ouliers = twt | twt \in DS[T_i] \wedge twt \notin c \forall c \in C$

  $Merging(C, Tracking)$

  $EnrollingIsolatedMembers(Ouliers, Tracking)$

  for *each* e *in* $Tracking$ **do**

    Determining the Existence and Decay timestamps of e **if** $t_{Decay} \neq Undefined$ **then**

      Remove e from $Tracking$ list

      Appending e into $Finish$ list

    end

  end

end

The data should be cleaned before any analysis process in order to improve performance, and we use the following steps to pre-process the collected micro-text corpus.

- **Normalizing**: All micro-texts are converted to lower-case characters. Each keyword is examined to reduce the number of repetitive character to a normal case that does not usually have any character repeated continuously more than 2 times. Extra spaces are also eliminated from sentences and between words.

- **Cleaning Micro-texts**: For the event detection issue, several special pieces of data that are embedded in the micro-text content are not necessary, such as embedded URL, symbols, SNS user mentions, and so on. First, the items are removed from the micro-text, and then we replace all of the embedded hash-tags to the corresponding keywords by removing the special leading symbol #.

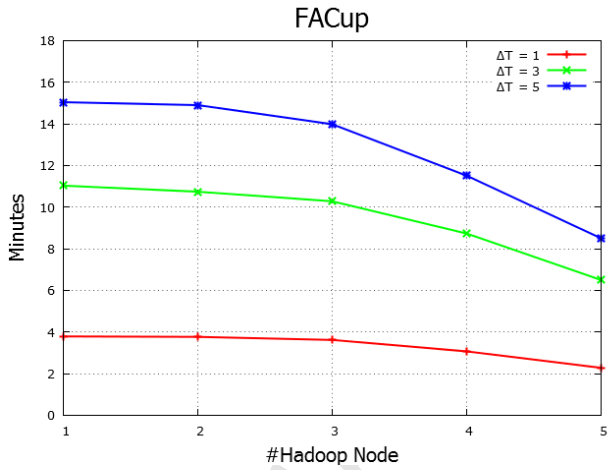
- **Filtering Trivial Keywords**: We determine the extracted keywords to be trivial if the signal power is not within a certain range of values. A coefficient $\alpha \geq 2$ is used to decide which keywords to be removed by applying a heuristic function to filter all keywords that do not have a power in the range from $\frac{(\alpha-1)\times \underset{\forall w}{Min}\, P_w + \underset{\forall w}{Avg}\, P_w}{\alpha}$ to $\frac{(\alpha-1)\times \underset{\forall w}{Max}\, P_w + \underset{\forall w}{Avg}\, P_w}{\alpha}$.

5.2. Evaluating

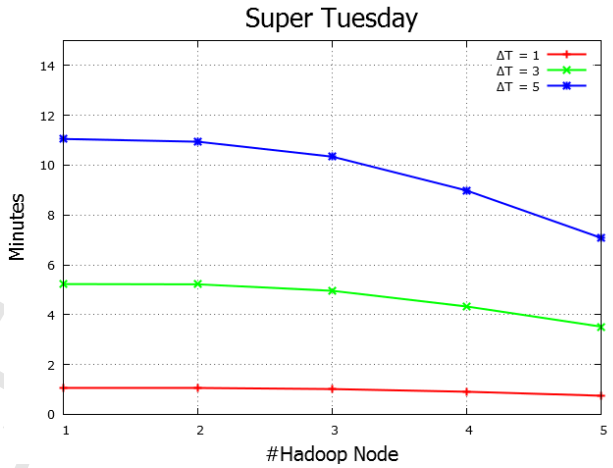
To evaluate the proposed method in terms of its accuracy, we reuse datasets introduced in [11]. Two datasets are selected for our research, including "FACup" and "Super Tuesday". The evaluation framework from [11] is also used for comparison with our event detection result, with the research result based on a list containing ground truth.

To evaluate the performance of our system, we configured a system using the Hadoop framework [21] to monitor the distributed processing units with a maximum number of nodes of 5 computers connected in a 100Mbps LAN. The computers have similar configurations with a 250 GB hard disk, an Intel Core 2

Duo 2.66Ghz CPU, 3 GB of RAM, and Ubuntu OS version 12.04. We predefined the number of time slots N = 128 for each sliding window and then simulate the discrete data stream from the given datasets by using three sampling intervals as 1, 3 and 5 minutes.



(a)



(b)

Figure 3: Average time for constructing signal of the a) "FACup", b) "Super Tuesday" dataset.

We recognize that the phase to construct the necessary signals is an intensive task that needs to be solved by improving the distributed computing and data storage capabilities. Figure 3 provides illustrations of the average time required to construct signals in one time slot with a ΔT of 1, 3, and 5 minutes. The "FACUP" dataset has an average number of tweets per one time slot corresponding to ΔT of 228, 648, 1080 tweets/slot each, and "Super Tuesday"has 228, 648, 1080 tweets/slot, respectively.

The number of tweets that are collected in each time slot also affects to the processing time length. The dataset "Super Tuesday" has a small average number of tweets per time slot because the tweets are distributed over a long time range, and the necessary time length to process each slot is less than that of the same task in the "FACup" dataset. Figure 3 also shows that a larger sampling ΔT interval requires the system to have more time to generate the signals for each keywords and tweet in one time slot. Therefore, we also need to consider the appropriate ΔT value that corresponds to the processing capability of the system to improve the use of the computing resources of the system.

Assuming that we have sufficient computing capability to generate all material signals for the event detection phase in the permitted period. We process the clustering phase on the "FACup" dataset by using one computer. The ground truth of the events that are observed during the final match in the FA Cup 2102 is collected from the BBC Sports website. The result is calculated using one computer with a sampling interval ΔT set to 3 minutes. In this case, the average of the processing time delay to being tracking the first tweet is of 222.96 seconds, and on average, the system confirms the event occurrence after 401.79 seconds relative to the real timestamps. The time delay is of 1.28 and 2.23 times the ΔT value. The reason such can be explained that, result of the signal construction task only being start at the end of each time slot, so the system has to wait until it sufficient data has arrived for processing.

Table 1: Average delay time for extracting events from the "FACup" dataset

ΔT (minute)	Measuring by seconds		Measuring according to ΔT	
	Start tracking	**Event Confirmation**	**Start tracking**	**Event Confirmation**
1 minute	71.56	207.60	1.19	3.46
3 minutes	222.96	401.79	1.24	2.23
5 minutes	361.98	462.00	1.21	1.54

Table 1 shows a comparison of the processing delay for the data stream simulation of the "FACup" dataset with three predefined sampling intervals. The amount of time needed to provide decisions according to two features is given. The results indicate an interesting conclusion in that the smaller value of ΔT makes the system more responsive to track and confirm events but actually requires more memory to store data, as discussed above. In the case of the "FACup" dataset, a smaller ΔT makes a system can only confirm the appearance of an event after a greater number of slots than the same process if a larger ΔT is used. This means that if we sample the data stream with a small sampling interval, it is not necessarily good to decrease the confirmation time length because the system is unable to collect a sufficient number of tweets over a short period of time to confirm the occurrences of such events. Hence, we need to consider to decide the value of ΔT depends on the context of each input dataset.

Each event in the given datasets is expressed as a set of meaningful keywords, and the results are evaluate through three features including the Topic Recall (TOP-REC), which is the percentage of ground truth topics that are successfully detected; the Keyword Precision (K-PREC), which is the percentage of keywords that are correctly detected over the total number of keywords for the topics that match any ground truth topic; and the Keyword Recall (K-REC),

which is the percentage of correctly detected keywords over the total number of keywords of the ground truth topics that match any candidate topic. The framework can be used to evaluate the accuracy of our event detection result, as shown in Table 2

Table 2: Event Detection Result.

Dataset	Feature	Our Method	BNGram	LDA
FACup	T-REC	**0.769**	**0.769**	0.692
	K-PREC	**0.453**	0.355	0.230
	K-REC	0.548	**0.587**	0.511
Super Tuesday	T-REC	0.455	**0.500**	0.182
	K-PREC	**0.652**	0.628	0.325
	K-REC	**0.714**	0.647	0.54

6. Conclusions

In this paper, we have presented a model to detect real-time events that draw public attention in a phenomenon where each SNS user plays a role as a social information sensor. The sensors are independent, but together share interesting facts that happen in their daily life. The events are indicated not only with content in the relative micro-texts that are shared between a group of friends, but also by how deep and fast news is diffused over the friendship network of the author. Our method uses common features that are related to user behavior to exchange interesting news on the Internet by combining it with a transformation method to reduce the complexity of data. The results of the experiment show that our approach has a high level of accuracy and efficiency to track abnormal phenomena in the social data stream.

The method is scalable and can be implemented to track real-time events

that are observed by multiple users independent of their location. This approach can work well by using only current observation data over a short range of time for a certain sliding window rather than monitoring the entire data stream.

6.1. Limitation

In this work, we have used a dataset that was introduced in prior research and consists of a tweet collection from Twitter. However, since the Twitter has updated its policies and old tweets have been deleted, the the corpus could not be completely downloaded. In addition, there is also a lack of information of the friendship network of users and relevant feedback for each tweet. This problem directly affects the precision of our detection results.

Although the results are encouraging, these have only been evaluated over a small-scale corpus in a simulation environment in a laboratory. Natural Language Processing techniques have still not been properly considered to improve the keyword extraction. Thus the original content of the tweets is often broken into single words, which probably leads to multiple variations that need to be analyzed.

In addition, the thresholds and coefficients were not set in a flexible manner to accommodate various kinds of data. In reality, each topic or corpus has a different data distribution that depends on the specific circumstances that are decided by the SNS users. This problem should therefore be reviewed in further detail to allow the system to self-adjust parameters by learning from a training dataset or from the dataset that is currently being analyzed.

6.2. Future work

For future work, we intend to evaluate this approach using a real-time data stream of an SNS like Twitter or Facebook with sufficient computing resources. The proposed method needs to be improved by investing more resources to increase the effectiveness of the detection result. The semantic relationship between the terms should be considered using a knowledge-base to identify a better combination function for Equation 5. A named-entity recognition method

or other NLP techniques are also needed to extract significant components from the content of micro-texts.

We also need to invest more time and effort to complete the technique to detect events on SNSs in real-time. In particularly, weak events that can be included in a dataset of various topics, which is often obscured by major events.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) with a grant funded by the Korean government (MSIP) (No. NRF-2014R1A2A2A05007154).

[1] S. Phuvipadawat, T. Murata, Breaking news detection and tracking in twitter, in: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology - Workshops, Toronto, Canada, August 31 - September 3, 2010, 2010, pp. 120–123.

[2] B. O'Connor, M. Krieger, D. Ahn, Tweetmotif: Exploratory search and topic summarization for twitter, in: Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010, 2010.

[3] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, J. Sperling, Twitterstand: news in tweets, in: 17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2009, November 4-6, 2009, Seattle, Washington, USA, Proceedings, 2009, pp. 42–51.

[4] G. P. C. Fung, J. X. Yu, P. S. Yu, H. Lu, Parameter free bursty events detection in text streams, in: Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005, 2005, pp. 181–192.

[5] H. Becker, M. Naaman, L. Gravano, Beyond trending topics: Real-world event identification on twitter, in: Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011, 2011.

[6] J. Yang, J. Leskovec, Patterns of temporal variation in online media, in: Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011, 2011, pp. 177–186.

[7] J. Lehmann, B. Gonçalves, J. J. Ramasco, C. Cattuto, Dynamical classes of collective attention in twitter, in: Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012, 2012, pp. 251–260.

[8] D. A. Shamma, L. Kennedy, E. F. Churchill, Peaks and persistence: modeling the shape of microblog conversations, in: Proceedings of the 2011 ACM Conference on Computer Supported Cooperative Work, CSCW 2011, Hangzhou, China, March 19-23, 2011, 2011, pp. 355–358.

[9] R. Li, K. H. Lei, R. Khadiwala, K. C. Chang, TEDAS: A twitter-based event detection and analysis system, in: IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012, 2012, pp. 1273–1276.

[10] S. Petrovic, M. Osborne, V. Lavrenko, Streaming first story detection with application to twitter, in: Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA, 2010, pp. 181–189.

[11] L. M. Aiello, G. Petkos, C. J. Martín, D. Corney, S. Papadopoulos, R. Skraba, A. Göker, I. Kompatsiaris, A. Jaimes, Sensing trending topics in twitter, Multimedia, IEEE Transactions on 15 (6) (2013) 1268–1282.

[12] J. G. Proakis, D. K. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications (4th Edition), 4th Edition, Prentice Hall, 2006.

[13] Q. He, K. Chang, E. Lim, Analyzing feature trajectories for event detection, in: SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, SIGIR '07, ACM, 2007, pp. 207–214.

[14] J. Weng, B. Lee, Event detection in twitter, in: Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011, 2011.

[15] T. Sakaki, M. Okazaki, Y. Matsuo, Earthquake shakes twitter users: real-time event detection by social sensors, in: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010, 2010, pp. 851–860.

[16] J. Mahmud, J. Nichols, C. Drews, Home location identification of twitter users, ACM Transactions on Intelligent Systems and Technology 5 (3) (2014) 47:1–47:21.

[17] G. G. Chowdhury, Introduction to Modern Information Retrieval, third edition Edition, Neal-Schuman Publishers, 2010.

[18] J. J. Jung, Ubiquitous conference management system for mobile recommendation services based on mobilizing social networks: a case study of u-conference, in: Expert Systems with Applications, 2011, pp. 12786–12790.

[19] E. International, Introducing json, Website, last checked: 2014-12-15 (2014).
URL `http://json.org`

[20] M. Ankerst, M. M. Breunig, H. Kriegel, J. Sander, OPTICS: ordering points to identify the clustering structure, in: SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA., 1999, pp. 49–60.

[21] T. White, Hadoop: The Definitive Guide, third edition Edition, O'REILLY, 2012.

***Biographies (Text)**

Author Biography

Dr. Duc T. Nguyen works for Faculty of Information Technology in Vietnam Maritime University since 2007. He received Ph.D. Degrees in Department of Computer Engineering of Yeungnam University (YU), Korea in 2015. Before to study in YU, he have received Master of Science Degree from Vietnamese National University in Hanoi. His research topic is knowledge engineering on social networks, he has released several publications related to analysis on Social Media, Event detections, Understanding Users' Behavior on SNS. He is interested in new technology, machine learning, intelligent computing, information system, and mobile application. He always welcomes any chance to understand and research more deeply into his trained profession in order to serve his teaching job and to apply his knowledge into practical production better.

Dr. Jai E. Jung is an Associate Professor in Chung-Ang University, Korea, since September 2014. Before joining CAU, he was an Assistant Professor in Yeungnam University, Korea since 2007. Also, He was a postdoctoral researcher in INRIA Rhone-Alpes, France in 2006, and a visiting scientist in Fraunhofer Institute (FIRST) in Berlin, Germany in 2004. His research topics are knowledge engineering on social networks by using many types of AI methodologies, e.g., data mining, machine learning, and logical reasoning. Recently, he have been working on intelligent schemes to understand various social dynamics in large scale social media (e.g., Twitter and Flickr).

***Biographies (Photograph)**
Click here to download high resolution image

***Biographies (Photograph)**
Click here to download high resolution image

Highlights

- The study proposes a real-time event detection method on Twitter.
- Frequency-based analytics has shown better performance on social streams.
- Online behavioral analysis on multiple users has been applied on social big data.