

RESEARCH

Open Access

# Real-time analysis application for identifying bursty local areas related to emergency topics

Tatsuhiko Sakai<sup>†</sup> and Keiichi Tamura<sup>\*†</sup>

## Abstract

Since social media started getting more attention from users on the Internet, social media has been one of the most important information source in the world. Especially, with the increasing popularity of social media, data posted on social media sites are rapidly becoming collective intelligence, which is a term used to refer to new media that is displacing traditional media. In this paper, we focus on geotagged tweets on the Twitter site. These geotagged tweets are referred to as georeferenced documents because they include not only a short text message, but also the documents' posting time and location. Many researchers have been tackling the development of new data mining techniques for georeferenced documents to identify and analyze emergency topics, such as natural disasters, weather, diseases, and other incidents. In particular, the utilization of geotagged tweets to identify and analyze natural disasters has received much attention from administrative agencies recently because some case studies have achieved compelling results. In this paper, we propose a novel real-time analysis application for identifying bursty local areas related to emergency topics. The aim of our new application is to provide new platforms that can identify and analyze the localities of emergency topics. The proposed application is composed of three core computational intelligence techniques: the Naive Bayes classifier technique, the spatiotemporal clustering technique, and the burst detection technique. Moreover, we have implemented two types of application interface: a Web application interface and an android application interface. To evaluate the proposed application, we have implemented a real-time weather observation system embedded the proposed application. we used actual crawling geotagged tweets posted on the Twitter site. The weather observation system successfully detected bursty local areas related to observed emergency weather topics.

**Keywords:** Spatiotemporal clustering; Density-based clustering; Social media; Naive Bayes; Burst detection

## Introduction

In recent years, social media has played a significant role as an alternative source of information (Kavanaugh et al. 2011; Yin et al. 2012). In particular, people actively transmit and collect information about emergency topics, such as natural disasters, weather, diseases, and other incidents (Miyabe et al. 2012; Vieweg et al. 2010). Enhancement of the utilization of social media for emergency management is one of the most interesting issues being discussed in public and governmental institutions. Therefore, a significant number of researchers have focused on the development of emergency topic and event detection via social media. This trend provides an opportunity for

addressing new challenges in many different application domains: how to detect where emergency topics occur and what they are going on.

In this study, we focus on geotagged tweets posted on the Twitter site. These geotagged tweets are referred to as georeferenced documents because they usually include not only a short text message, but also the documents' posting time and location. Users on the Twitter site are referred to as social sensors and geotagged tweets as sensor data observed by the social sensors. Some of studies that focused on these geotagged tweets are as follows. Sakaki et al. (2010) focused on tweets regarding typhoons and earthquakes using associated geographic information to estimate typhoon trajectories and earthquake epicenters using dense regions. Ozdakis et al. (2013) also proposed a method that estimated the geographical location of events in the case of earthquakes reported on Twitter.

\*Correspondence: ktamura@hiroshima-cu.ac.jp

<sup>†</sup>Equal contributors

Graduate School of Information Sciences, Hiroshima City University, 3-4-1, Ozuka-Higashi, Asa-Minami-Ku, 731-3194 Hiroshima, Japan

It is of value to people interested in a certain topic to observe dense areas where many georeferenced documents related to the topic are located. In this paper, these dense areas are referred to as bursty local areas related to the topic. For example, Tamura and Ichimura (2013) proposed a novel density-based spatiotemporal clustering algorithm that can extract spatially and temporally separated spatial clusters in georeferenced documents. They proposed the  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm; the experimental results indicate that their proposed algorithm can extract bursty local areas in a set of geotagged tweets, including keywords related to a topic related to weather topics.

In this paper, we propose a novel real-time analysis application for identifying bursty local areas related to emergency topics. The aim of our new application is to provide new platforms that can identify and analyze the localities of emergency topics. The proposed application is composed of three core computational intelligence techniques: the Naive Bayes classifier technique, the spatiotemporal clustering technique, and the burst detection technique. The  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm is a useful algorithm for extracting bursty local areas; however, two functional issues remain unresolved. One issue is that the  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm does not support real-time extraction. In Tamura and Ichimura (2013), only a batch clustering algorithm was proposed. The second issue is that the proposed algorithm is based on keywords. Therefore, relevant georeferenced documents are extracted if they include an observed keyword, not an observed topic; and this causes error extraction.

The main characteristics of our application are as follows.

- To extract georeferenced documents including emergency topics as relevant georeferenced documents, our application utilizes the Naive Bayes classifier Manning et al. (2008) technique. As compared to that proposed in Tamura and Ichimura (2013), the proposed application can detect spatiotemporal clusters as bursty local areas with more sensitivity. For example, suppose that there are two sentences, "I saw a snow monkey" and "It is snowing heavily". These two sentences include a keyword "snow". The first one is not related to weather topic "snow". Our application is topic-based; therefore the first one is not extracted as a relevant georeferenced document.
- To identify  $(\epsilon, \tau)$ -density-based spatiotemporal clusters in real time, an incremental algorithm for  $(\epsilon, \tau)$ -density-based spatiotemporal clustering is used. The target topics of this study are emergency

topics; therefore we need to identify local bursty areas in real time. In Tamura and Ichimura (2013), only a batch clustering algorithm was proposed. In contrast, the incremental algorithm extracts  $(\epsilon, \tau)$ -density-based spatiotemporal clusters on the arrival of each relevant georeferenced document.

- To identify the burstiness of extracted bursty local areas, we integrate location-based burst detection techniques Tamura and Kitakami (2013) in the proposed application. Burst Kleinberg (2002) is one of the simplest but most effective criterion to measure how hot topics are. The traditional time-based burst detection does not work well for this study because the emergency topics appear in local area. The location-based burst detection techniques can detect the localities of burstiness of emergency topics.

The rest of this paper is organized as follows. In Section Related work, we briefly describe related work. In Section  $(\epsilon, \tau)$ -Density-based spatiotemporal clustering, the density-based clustering algorithm and the  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm are reviewed. In Section The proposed application, we propose our application and describe the details of the core intelligent computational techniques implemented in it. In Section Experimental result, the experiments for the evaluation of the proposed application is reported. In Section Conclusion, we conclude this paper.

## Related work

In the era of big data, we are witnessing the rapid growth of a new type of information source. Social media has been noticed by a significant number of people; today, we obtain information instantaneously about emergency topics that surround us. In particular, tweets from one of the most widely used micro-blogging services have been involved in many different application domains Java et al. (2007). Tweets include not only rubbish messages transmitted between users, but also comments about and content for social topics and events Jansen et al. (2009). Twitter users are referred to as sensors observing the world, and their tweets as sensor data.

Recently, a huge amount of geotagged tweets are posted because of the popularity of geomobile applications on smartphones. These geotagged tweets are useful for extracting local topics and events (Abdelhaq et al. 2013; Hiruta et al. 2012; Hong et al. 2012; Musleh 2014). In particular, geotagged tweet have been expected to be utilized for analyzing emergency topics like natural disasters (Mandel et al. 2012) and epidemic (Hwang et al. 2013).

The most related work is Aramaki et al. (2011) which were conducted by Aramaki et al. They proposed a novel method for detecting influenza epidemics using tweets; their method utilized classifiers, such as support vector

machine (SVM) and Naive Bayes to extract the tweets that included topics about influenza. Their geocoding technique was used to map each tweet to a region in Japan. Moreover, their proposed system visualized the increase and decrease in the number of related tweets in each region. As indicated above, their approach is more closely related to our work than any other; however, the method cannot identify bursty areas in detail. Our proposed method can detect large regions like prefectures.

Murakami et al. analyzed tweets about the 2011 Tohoku earthquake and tsunami Murakami and Nasukawa (2012). They presented the analysis results of social media data using text mining tools. Marcus et al. (2011) developed a visualizing system called Twitinfo. Their system is suitable for analyzing global topics and their time change. Moreover, the system visualized the sentiment of each topic. Karimi et al. (2013) proposed classifying methods that can identify high-value tweets related to disasters. Their method showed a good performance; however, they proposed only the classifying methods.

Kaneko et al. (2013) proposed a novel method for detecting images of events such as the cherry-blossom festival using tweets that included geo-tags and images. First, their method checks the number of tweets posted on a day and the preceding day and also extracts keywords from the posts. Next, it searches and collects images of tweets using these keywords. Moreover, their method performs clustering using the feature value of the collected images and selects a representative image as an event; however, their method cannot detect bursty areas in detail, because it checks only the changes from one day to the next.

Typhoon Real Time Watcher (2014) was developed recently. This system can observe and track social typhoon reports about their rain, wind, and damages, which are provided by a weather forecast company. This system utilized social reports manually selected; therefore, the reliability is assured. The system is good tool for analyzing typhoons; however, it is difficult to obtain real-time situation because it utilizes only social typhoon reports provided by a weather forecast company. NTT DoCoMo proposed an application called Geographical TimeLine that uses its own auto GPS system and Twitter. The application can detect in real time a place where many people are using its auto GPS system. Moreover, the application can analyze events that occur by exploiting Twitter; however, it requires an auto GPS system.

### $(\epsilon, \tau)$ -Density-based spatiotemporal clustering

This section reviews the density-based clustering framework and the  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm proposed in Tamura and Ichimura (2013).

The  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm is the improved version of DBSCAN algorithm Ester et al. (1996).

### Density-based spatiotemporal criteria

The  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm is based on the density-based spatial clustering algorithm Sander et al. (1998). In the density-based spatial clustering algorithm, spatial clusters are dense areas that are separated from areas of lower density. In other words, areas with high densities of data points can be considered spatial clusters, whereas those with low density cannot. The key concept underpinning the use of the density-based spatial clustering algorithm indicates that, for each data point within a spatial cluster, the neighborhood of a user-defined radius must contain at least a minimum number of points; that is, the density in the neighborhood must exceed some predefined threshold.

The algorithm that has affected the density-based spatial clustering algorithm is the DBSCAN algorithm, which was first introduced by Ester et al. (1996). The  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm Tamura and Ichimura (2013) is a natural extension of DBSCAN. DBSCAN utilizes  $\epsilon$ -neighborhood density and recognizes areas in which densities are higher than in other areas. However, it does not consider temporal changes. By contrast, the  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm extracts  $(\epsilon, \tau)$ -density-based spatiotemporal clusters that are both temporally and spatially-separated from other spatial clusters.

In the  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm, areas with high spatial-/temporal-densities of data points can be considered spatiotemporal clusters, whereas those with low spatial-/temporal-densities cannot. To extract spatiotemporal clusters, we extend the definitions of density in DBSCAN. Suppose that an emergency topic is getting more attention in a local area. Users in the local area start posting geotagged tweets related to the emergency topic; therefore, if we detect high dense regions in which there are many geotagged tweets, we can analyze local bursty areas where emergency topic is getting more attention from local people.

### Definitions

The  $(\epsilon, \tau)$ -density-based neighborhood, which indicates the density of the neighborhood of a georeferenced document, is defined as follows.

**Definition 1** ( $(\epsilon, \tau)$ -density-based neighborhood). The  $(\epsilon, \tau)$ -density-based neighborhood of a georeferenced document  $gdp$ , which is denoted by  $N_{(\epsilon, \tau)}(gdp)$ , is defined as

$$N_{(\epsilon, \tau)}(gdp) = \{gdp \in GD \mid \text{dist}(gdp, gdq) \leq \epsilon \text{ and } \text{iat}(gdp, gdq) \leq \tau\}, \quad (1)$$

where the function *dist* returns the distance between georeferenced documents *gdp* and *gdq*, and the function *iat* returns the interarrival time between them.

Figure 1 shows an example of an  $(\epsilon, \tau)$ -density-based neighborhood. In DBSCAN, the neighborhood of document *gdp* is a set of documents that exist within  $\epsilon$  from *gdp*. In the left-hand side of Figure 1, there are four documents in the neighborhood of *gdp*. Conversely, the  $(\epsilon, \tau)$ -density-based neighborhood of *gdp* is a set of documents that exist within  $\epsilon$  from *gdp*, where each document in the  $(\epsilon, \tau)$ -density-based neighborhood is posted in  $\tau$  before or after the posted time of document *gdp*. The right-hand side of Figure 1 shows the example of the  $(\epsilon, \tau)$ -density-based neighborhood. In this example, there are three documents,  $N_{(\epsilon, \tau)}(gdp) = \{gd_2, gd_3, gd_4\}$ . Document *gd*<sub>1</sub> is within  $\epsilon$  from document *gdp*; however, it is not in  $N_{(\epsilon, \tau)}(gdp)$ , because it is not posted in  $\tau$  before or after the posted time of document *gdp*.

**Definition 2** (Core and Border). A georeferenced document *gdp* is called a core georeferenced document if there is at least a minimum number of georeferenced documents, *MinGDoc*, in the  $(\epsilon, \tau)$ -density-based neighborhood  $N_{(\epsilon, \tau)}(gdp)$  of that georeferenced document ( $|N_{(\epsilon, \tau)}(gdp)| \geq MinGDoc$ ). Otherwise, ( $|N_{(\epsilon, \tau)}(gdp)| \leq MinGDoc$ ), a georeferenced document *gdp* is called a border georeferenced document.

Suppose that *MinGDoc* is set to three. In the left-hand side of Figure 2, *gdp* is a core georeferenced document, because  $|N_{(\epsilon, \tau)}(gdp)|$  is three. In the right-hand side of Figure 2, *gdp* is not a core georeferenced document, because  $|N_{(\epsilon, \tau)}(gdp)|$  is three.

**Definition 3** ( $(\epsilon, \tau)$ -density-based directly reachable). Suppose that the georeferenced document *gdq* is in the  $(\epsilon, \tau)$ -density-based neighborhood of georeferenced

document *gdp*. If the number of georeferenced documents in the  $(\epsilon, \tau)$ -density-based neighborhood of georeferenced document *gdp* is greater than or equal to *MinGDoc*, i.e., is  $|N_{(\epsilon, \tau)}(gdp)| \geq MinGDoc$ , document *gdq* is  $(\epsilon, \tau)$ -density-based directly reachable from *gdp*. In other words, georeferenced documents in the  $(\epsilon, \tau)$ -density-based neighborhood of a core georeferenced document are  $(\epsilon, \tau)$ -density-based directly reachable from the core georeferenced document.

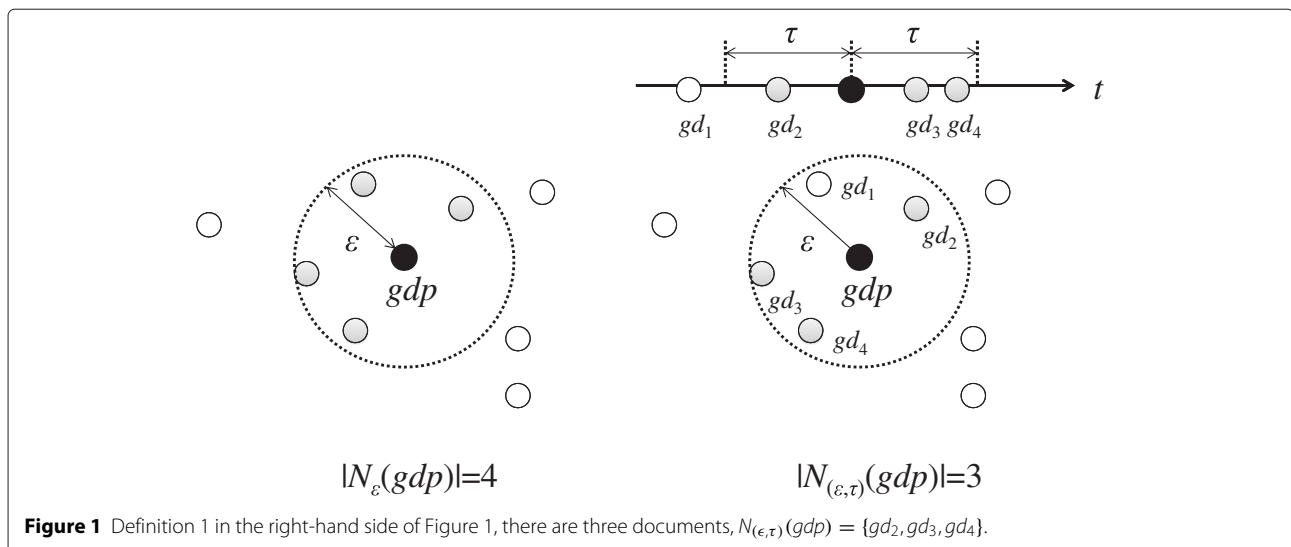
On the left-hand side of Figure 2, *gdp* is a core georeferenced document, because  $N_{(\epsilon, \tau)}(dp) \geq MinGDoc$ . Then, *gd*<sub>1</sub>, *gd*<sub>2</sub> and *gd*<sub>4</sub> are in the  $(\epsilon, \tau)$ -density-based neighborhood of document *gdp*. These three georeferenced documents are  $(\epsilon, \tau)$ -density-based directly reachable from *gdp*.

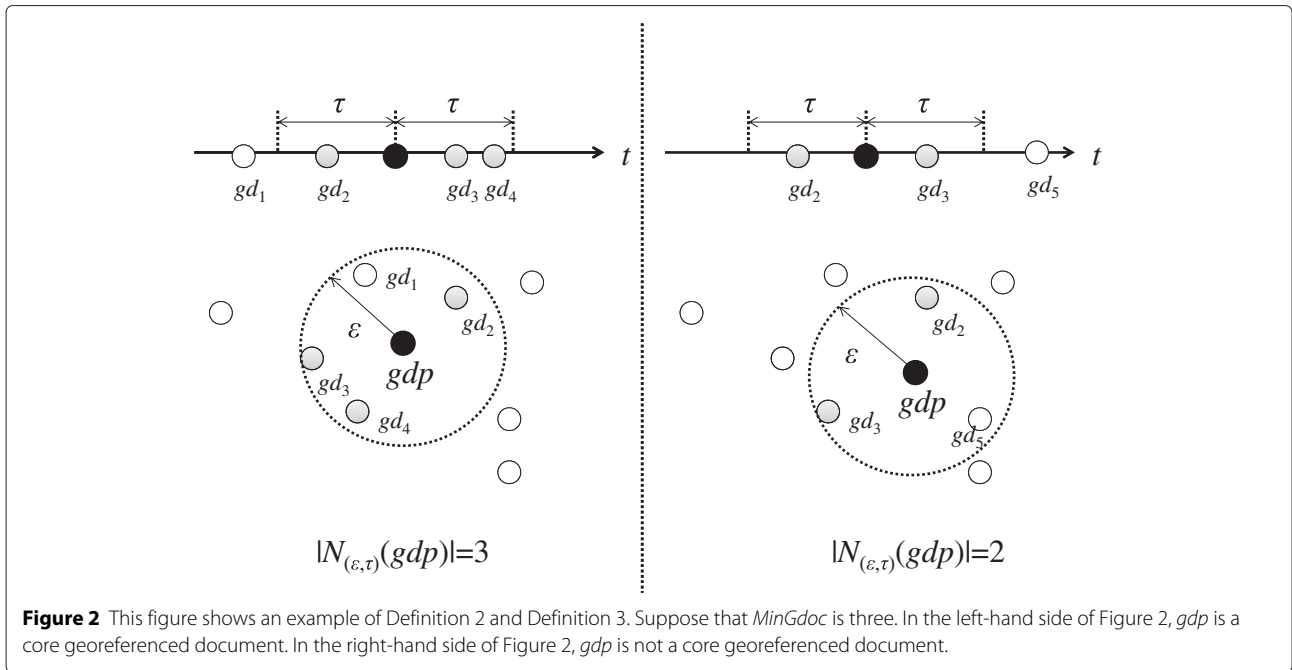
**Definition 4** ( $(\epsilon, \tau)$ -density-based reachable). Suppose that there is a georeferenced document sequence (*gdp*<sub>1</sub>, *gdp*<sub>2</sub>, ..., *gdp*<sub>*n*</sub>) and the *i* + 1-th georeferenced document *gdp*<sub>*i*+1</sub> is  $(\epsilon, \tau)$ -density-based directly reachable from the *i*-th georeferenced document *gdp*<sub>*i*</sub>. Georeferenced document *gdp*<sub>*n*</sub> is  $(\epsilon, \tau)$ -density-based reachable from georeferenced document *gdp*<sub>1</sub>.

Figure 3 shows an example of definition 4. In a georeferenced document sequence (*gdp*<sub>1</sub>, *gdp*<sub>2</sub>, ..., *gdp*<sub>5</sub>), the *i* + 1-th georeferenced document *gdp*<sub>*i*+1</sub> is  $(\epsilon, \tau)$ -density-based directly reachable from *gdp*<sub>*i*</sub>. Therefore, *dp*<sub>5</sub> is  $(\epsilon, \tau)$ -density-based reachable from *gdp*<sub>1</sub>.

**Definition 5** ( $(\epsilon, \tau)$ -density-based connected). Suppose that georeferenced documents *gdp* and *gdq* are  $(\epsilon, \tau)$ -density-based reachable from a georeferenced document *gdo*, which is an arbitrary georeferenced document. If  $|N_{(\epsilon, \tau)}(gdo)| \geq MinGDoc$ , we denote that *gdp* is  $(\epsilon, \tau)$ -density-based connected to *gdq*.

Figure 4 shows an example of definition 5. In this example, *gdp*<sub>6</sub> is  $(\epsilon, \tau)$ -density-based connected to *gdp*<sub>5</sub>,





because  $gdp_5$  is  $(\epsilon, \tau)$ -density-based reachable from  $gdp_2$  and  $gdp_6$  is  $(\epsilon, \tau)$ -density-based reachable from  $gdp_2$ .

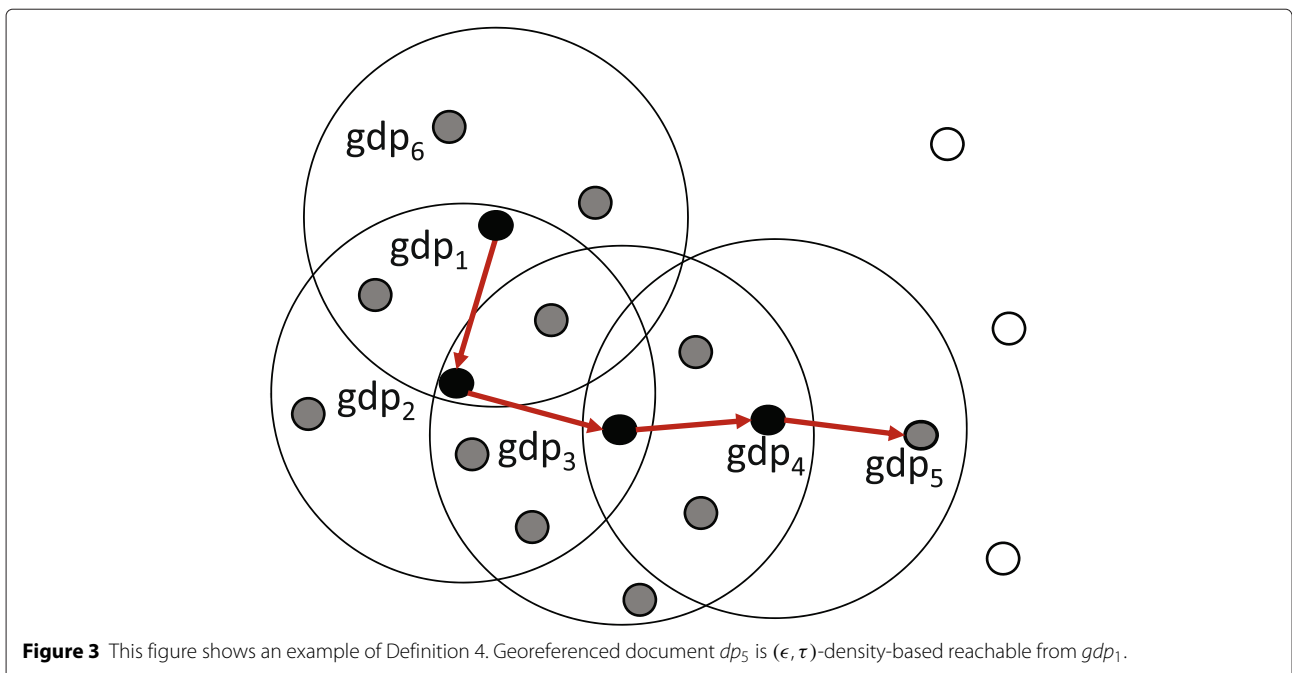
**$(\epsilon, \tau)$ -Density-based spatiotemporal cluster**

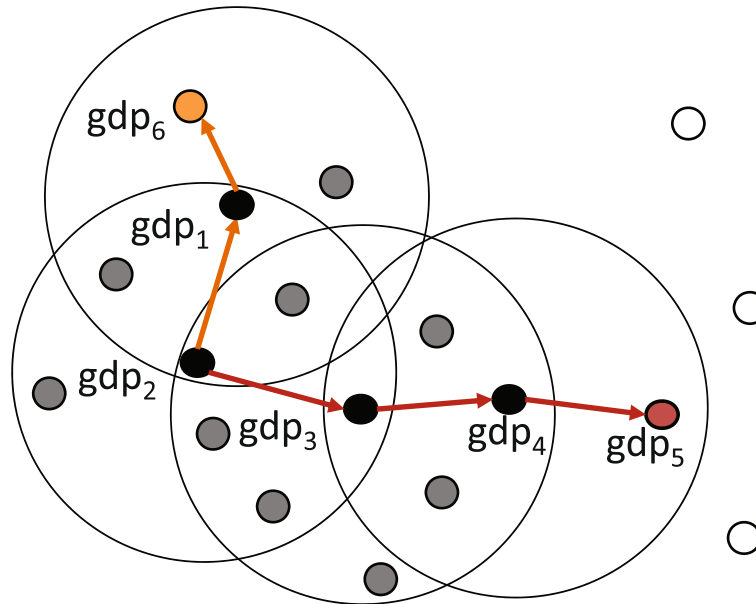
An  $(\epsilon, \tau)$ -density-based spatiotemporal cluster consists of two types of georeferenced document: core georeferenced documents, which are mutually  $(\epsilon, \tau)$ -density-based reachable; and border georeferenced documents, which are  $(\epsilon, \tau)$ -density-based directly reachable from the

core georeferenced documents. An  $(\epsilon, \tau)$ -density-based spatiotemporal cluster is defined as follows.

An  $(\epsilon, \tau)$ -density-based spatiotemporal cluster ( $DSC$ ) in a georeferenced document set  $GD$  satisfies the following restrictions:

- (1)  $\forall gdp, gdq \in GD$ , if and only if  $gdp \in DSC$  and  $gdq$  is  $(\epsilon, \tau)$ -density-based reachable from  $gdp$ , and  $gdq$  is also in  $DSC$ .





**Figure 4** This figure shows an example of Definition 5. Georeferenced document  $gdp_6$  is  $(\epsilon, \tau)$ -density-based connected to  $gdp_5$ .

(2)  $\forall gdp, gdq \in DSC, gdp$  is  $(\epsilon, \tau)$ -density-based connected to  $gdq$ .

Even if  $gdp$  and  $gdq$  are border georeferenced documents,  $gdp$  and  $gdq$  are in a same  $(\epsilon, \tau)$ -density-based spatiotemporal cluster if  $gdp$  is  $(\epsilon, \tau)$ -density-based connected to document  $gdq$ .

**Algorithm**

Algorithm 1 describes the batch algorithm for  $(\epsilon, \tau)$ -density-based spatiotemporal clustering. In this algorithm, for each georeferenced document  $gdp$  in  $GD$ , the function **IsClustered** checks whether document  $gdp$  is already assigned to a spatiotemporal cluster. Then, the  $(\epsilon, \tau)$ -density-based neighborhood of document  $gdp$  is obtained using the function **GetNeighborhood**. If georeferenced document  $gdp$  is a core document according to Definition 2, it is assigned to a new spatiotemporal cluster, and all the neighbors are queued to  $Q$  for further processing. The processing and assignment of georeferenced documents to the current spatiotemporal cluster continues until the queue is empty. The next georeferenced document is dequeued from queue  $Q$ . If the dequeued georeferenced document is not already assigned to the current spatiotemporal cluster, it is assigned to the current spatiotemporal cluster. Then, if the dequeued document is a core document, the georeferenced documents in the  $(\epsilon, \tau)$ -density-based neighborhood of the dequeued georeferenced document are queued in queue  $Q$  using the function **EnNniqueQueue**, which places the input georeferenced documents into queue  $Q$  if they are not already in queue  $Q$ .

**Algorithm 1:**  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm

**input** :  $GD$  - a set of georeferenced document,  $\epsilon$  - neighborhood radius,  $\tau$  - interarrival time,  $MinGDoc$  is threshold value  
**output**:  $STC$  - set of spatiotemporal clusters

```

cid ← 1;
STC ← φ;
for i ← 1 to |GD| do
    pd ← gdi ∈ GD;
    if IsClustered(gdp) == false then
        N ← GetNeighborhood(gdp, ε, τ);
        if |N| ≥ MinGDoc then
            stccid ← MakeNewCluster(cid, gdp);
            cid ← cid + 1;
            EnQueue(Q, N);
            while Q is not empty do
                pq ← DeQueue(Q);
                stccid ← stccid ∪ pq;
                N ← GetNeighborhood(gdq, ε, τ);
                if |N| ≥ MinGDoc then
                    EnNniqueQueue(Q, N);
                end if
            end while
            STC ← STC ∪ stccid;
        end if
    end if
end for
return STC;

```

### The proposed application

In this section, we propose a novel real-time analysis application for identifying bursty local areas related to emergency topics. A system overview and the system process are presented.

#### Aim

The aim of developing our application is to provide a platform that can analyze bursty local areas related to emergency topics in real time. Suppose that it is raining in area “A”. As the rain becomes heavily in area “A”, geotagged tweets related to the topic “rain” may be posted in the area. At the same time, the density of posted geotagged tweets become greater than usual. If the density areas are extracted, we can detect bursty local areas related to the emergency topic.

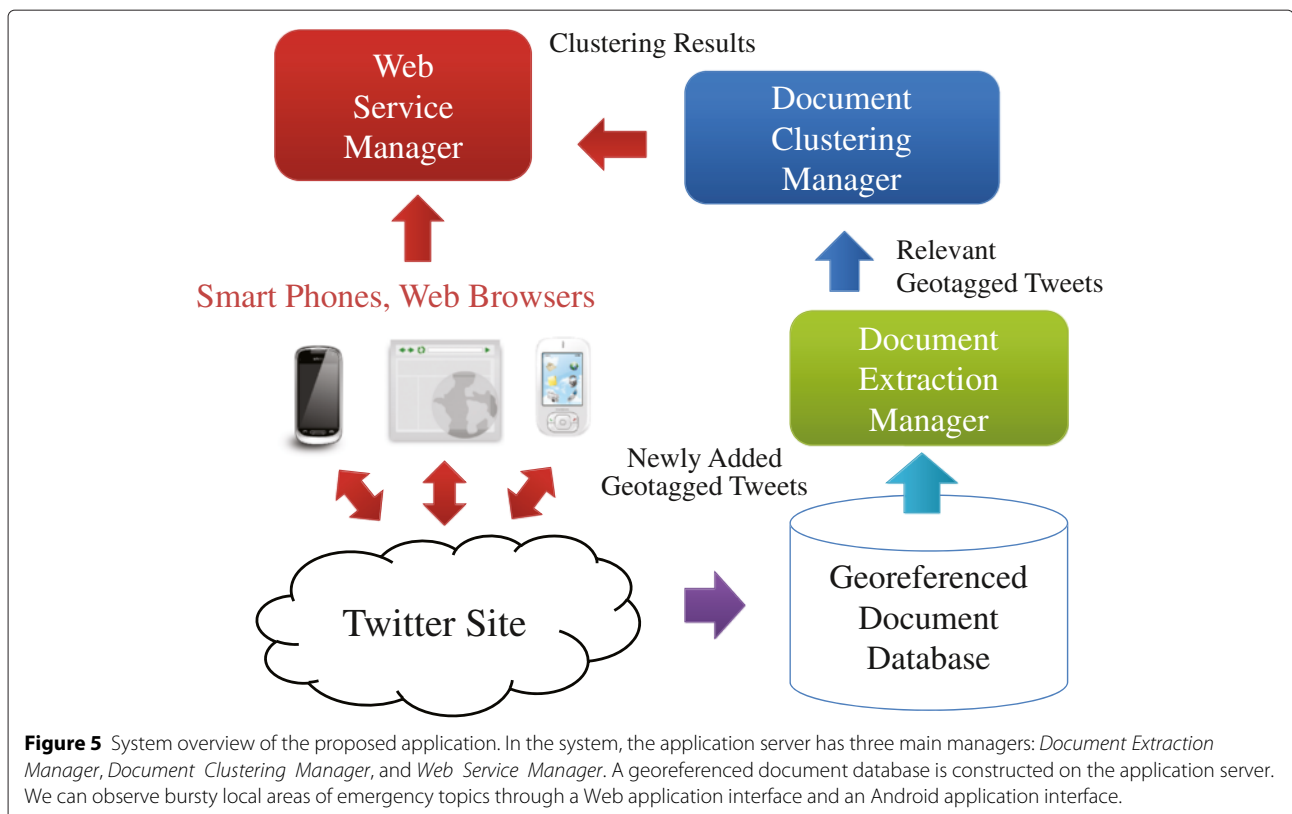
#### System overview

Figure 5 shows an overview of the system for the proposed application. In the system, the application server has three main managers: *Document Extraction Manager*, *Document Clustering Manager*, and *Web Service Manager*. We can observe bursty local areas of emergency topics through a Web application and an Android application.

Our system has a georeferenced document database that contains geotagged tweets crawled by the Twitter site. Let  $gd_i$  denote the  $i$ -th georeferenced document in  $GDS = \{gd_1, \dots, gd_n\}$ ; then,  $gd_i$  consists of three items:  $gd_i = \langle text_i, pt_i, pl_i \rangle$ , where  $text_i$  is the content (e.g., title, short text message, and tags),  $pt_i$ , which is the time when the georeferenced document was posted, and  $pl_i$ , which is the location where  $gd_i$  was posted or is located (e.g., latitude and longitude).

Each georeferenced document arrives step by step. The following steps are executed on the application server.

1. *Document Extraction Manager* fetches a georeferenced document, which is newly inserted in the georeferenced document database.
2. *Document Extraction Manager* classifies the fetched georeferenced document  $gd_i$  using a Naive Bayes classifier. If and only if  $gd_i$  is classified to “positive” class, which means  $gd_i$  is related to an observed emergency topic, go to the next step.
3. *Document Clustering Manager* executes the incremental algorithm for extracting  $(\epsilon, \tau)$ -density-based spatiotemporal clustering, for which there are two input data:  $gd_i$  and a set of current extracted  $(\epsilon, \tau)$ -density-based spatiotemporal clusters.



4. For each  $(\epsilon, \tau)$ -density-based spatiotemporal cluster, the burstiness of the cluster is calculated.
5. *Web Service Manager* provides Web-based application interfaces to access information about extracted bursty local areas.

Our system is not dependence on any language. For, example, the Naive Bays classifier can be conducted if any morphological analysis tool is provided. Moreover, the density-based spatiotemporal algorithm is without dependence on language. Therefore, we can extend the proposed system to another language if we modified the Web-based Interfaces and the Android applications in the language.

#### Naive Bayes classifier

The proposed application requires that georeferenced documents related to an observed emergency topic are extracted. Georeferenced documents including the observed emergency topic contain many kinds of keyword. Therefore, a keyword-based search is not effective for extraction. For example, suppose that an observed emergency topic is “rain”. Sequences “It is raining” and “It could rain this weekend” include the keyword “rain”; but, they have different topics. In this case, only “It is raining” is extracted as a relevant georeferenced document related to the topic “rain”.

To satisfy this requirement, in *Document Extraction Manager*, the Naive Bayes classifier technique is utilized in order to extract georeferenced documents. A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes’ theorem, which is based Bayesian statistics with naive independence assumptions. *Document Extraction Manager* classifies geotagged tweets as either “positive” class or “negative” class manually, where “positive” class is related to the observed emergency topic and “negative” class is not. Georeferenced documents in the “positive” class are the relevant georeferenced documents.

In this study, the Naive Bayes classifier is based on keywords in text data included in georeferenced documents. Let  $CLASS = \{positive, negative\}$  be a set of classes. The posterior probability that is the georeferenced document  $gd$  belongs the class  $class \in CLASS$  is

$$P(class|gd) = \frac{P(class)P(gd|class)}{P(gd)} \propto P(class)P(gd|class), \quad (2)$$

where  $P(class)$  is the prior probability of  $class$  and  $P(gd|class)$  is the likelihood.

The Naive Bayes classifier requires a training data set including multiple georeferenced documents that are classified in one class in  $CLASS$ . Let a training data set  $TDS$  be  $TDS = \{(tgd_1, c_1), (tgd_2, c_2), \dots, (tgd_m, c_m)\}$ , where  $c_i \in$

$CLASS$ . A set of all words in  $class$  is denoted by  $W_{class} = \{word_1, word_2, \dots, word_l\}$ .

The georeferenced  $gd$  represents a bag-of-words  $word_1, word_2, \dots, word_{numw(gd)}$ , where let  $numw(gd)$  be the number of words in  $gd$ .

$$\begin{aligned} P(gd|class) &= P(word_1 \wedge word_2 \wedge \dots \wedge word_k | class) \\ &= \prod_i^{numw(gd)} P(word_i | class) \end{aligned} \quad (3)$$

where,  $P(word_i | class)$  is the probability that  $word_i$  occurs in  $class$ .  $P(word_i | class)$  is defined as

$$P(word_i | class) = \frac{OW(word_i, class) + 1}{\sum_{j=1}^{|W_{class}|} (OW(word_j, class) + 1)}, \quad (4)$$

where  $OW(word_i, class)$  is the number of occurrence of  $word_i$  in  $class$ .

The assigned class of  $gd$  denoted by  $c_{gd} \in CLASS$  is determined by finding the maximum posterior probability.

$$\begin{aligned} c_{gd} &= \arg \max_{class} P(class|gd) \\ &= \arg \max_{class} \left( P(class) \prod_i^{numw(gd)} P(word_i | class) \right) \end{aligned} \quad (5)$$

#### Incremental algorithm

In the incremental algorithm, the algorithm updates the states of the extracted spatiotemporal clusters and extracts new spatiotemporal clusters every time a georeferenced document is added. Algorithm 2 describes the incremental  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm, which extracts  $(\epsilon, \tau)$ -density-based spatiotemporal clusters based on every georeferenced document that arrives for real-time extraction. There are two features in the incremental  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm: limited re-clustering and merging.

When a georeferenced document is added to the georeferenced documents, existing  $(\epsilon, \tau)$ -density-based spatiotemporal clusters must be updated; but the added georeferenced document affects only its  $(\epsilon, \tau)$ -density-based neighborhood within  $\tau$  directly. Function **GetRecentData**( $gd$ ) returns  $gd$ 's  $(\epsilon, \tau)$ -density-based neighborhood within  $\tau$ . After the  $(\epsilon, \tau)$ -density-based neighborhood is extracted to generate seeds and these seed georeferenced documents are re-clustered again.

In the incremental algorithm, during re-clustering, some  $(\epsilon, \tau)$ -density-based spatiotemporal clusters need to be appended to other  $(\epsilon, \tau)$ -density-based spatiotemporal clusters. Suppose that  $(\epsilon, \tau)$ -density-based spatiotemporal cluster  $stc$  is expanding. If a core georeferenced



**Algorithm 2:** Incremental  $(\epsilon, \tau)$ -density-based spatiotemporal clustering algorithm

---

```

input :  $gd$  - a newly input georeferenced document,
          $GD$  - a set of georeferenced,  $CSTC$  - a set of
         extracted spatiotemporal clusters,  $\epsilon$  -
         user-specified value,  $\tau$  - user-specified value,
          $MinGDoc$  - the minimum number of
         georeferenced document
output:  $NSTC$  - a set of updated spatiotemporal
         clusters

 $NSTC \leftarrow CSTC$ ;
 $RD \leftarrow \text{GetRecentData}(gd, \tau, GD)$ ;
for  $i \leftarrow 1$  to  $|RD|$  do
   $pd \leftarrow rd_i \in RD$ ;
   $N \leftarrow \text{GetNeighborhood}(pd, \epsilon, \tau)$ ;
  if  $|N| \geq MinGDoc$  then
    if  $\text{IsClustered}(pd) == \text{false}$  then
       $stc \leftarrow \text{MakeNewCluster}(cid, pd)$ ;
    end if
    else
       $stc \leftarrow \text{GetCluster}(pd, NSTC)$ ;
    end if
     $\text{EnQueue}(Q, N)$ ;
    while  $Q$  is not empty do
       $gd \leftarrow \text{DeQueue}(Q)$ ;
      if  $\text{IsClustered}(gd) == \text{true}$  then
         $N \leftarrow \text{GetNeighborhood}(gd, \epsilon, \tau)$ ;
        if  $|N| \geq MinGDoc$  then
           $stc' \leftarrow$ 
             $\text{GetCluster}(gd, NSTC)$ ;
           $stc \leftarrow$ 
             $\text{AppendClusters}(stc, stc')$ ;
        end if
      end if
    else
       $stc \leftarrow stc \cup gd$ ;
       $N \leftarrow \text{GetNeighborhood}(gd, \epsilon, \tau)$ ;
      if  $|N| \geq MinGDoc$  then
         $\text{EnNiqueQueue}(Q, N)$ ;
      end if
    end if
  end while
   $NSTC \leftarrow NSTC \cup stc$ ;
end if
end for
return  $NSTC$ ;

```

---

document in  $stc$  includes a georeferenced document, which is clustered in  $stc'$ ,  $stc'$  is appended to  $stc$ . Function **AppendClusters** appends two spatiotemporal clusters and return a appended spatiotemporal cluster.

**Burst detection**

To identify the burstiness of extracted areas, we integrate location-based burst detection techniques Tamura and Kitakami (2013) in the proposed algorithm. We extended the location-based burst detection algorithm for detecting the burstiness of  $(\epsilon, \tau)$ -density-based spatiotemporal clusters. In the location-based burst detection algorithm, there are two sequences, that of the number of georeferenced documents and that of the number of relevant georeferenced documents.

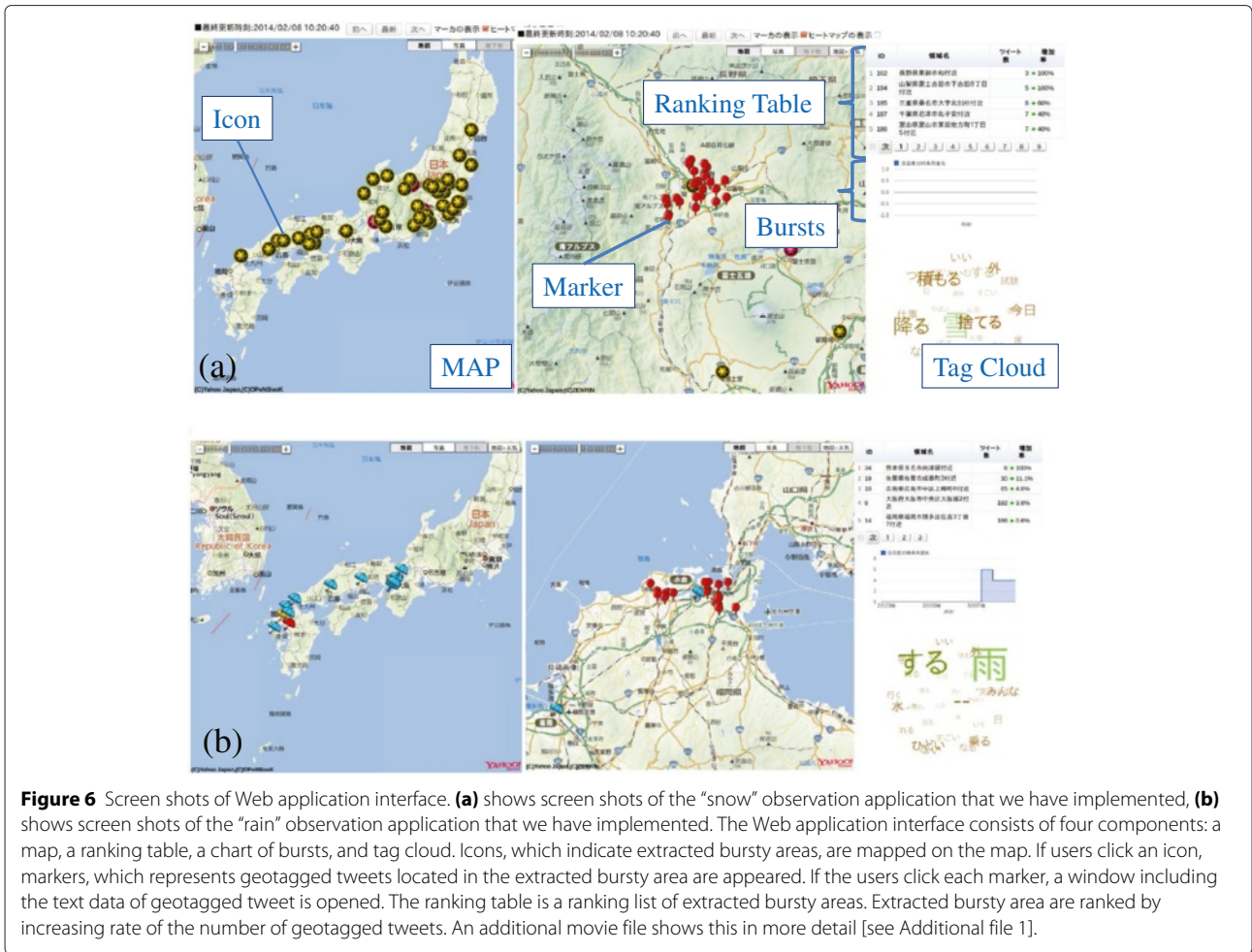
In this study, a sequence of the number of all the relevant georeferenced documents is referred to as the sequence of georeferenced documents and a sequence of the number of relevant georeferenced documents in each  $(\epsilon, \tau)$ -density-based spatiotemporal cluster is referred to as the sequence of relevant georeferenced documents. Moreover, the influence rates of georeferenced documents gradually decrease according to distance from the center of each cluster.

**Application interfaces**

Two application interfaces are used for accessing information on bursty local areas related to emergency topics: a Web application interface and an android application interface. These two types of application interfaces access a application server and users can analyze bursty local areas through the interfaces. There is a geographical map on the Web application interface and the android application interface. Bursty local areas mapped on the geographical map, and each geotagged tweets in extracted spatiotemporal clusters are shown in the geographical map. Moreover, we can watch posted image with geotagged tweets.

We implemented a real-time weather observation system embedded with the proposed application. The real-time weather observation system provides the functions that show bursty local areas in which the weather is receiving more attention from many people located in these local areas. We can observe how observed weather is going on in bursty local areas through this system. The weather, such as, rain, snow, and wind cause a severe natural disaster; therefore, this system contributes an analysis of the weather forecast.

Figures 6 and 7 show screen shots of the Web application interface and the Android application interface to the real-time weather observation system, which are implemented by us. Figure 6 (a) shows screen shots on February 8, 2014. It snowed heavily in Japan; especially, the Tokyo metropolitan region and Koshin region had heavy snow on February 8, 2014. The icons of snow crystal indicates extracted bursty local areas. Through the system, we can know what weather is going on in Japan. Figure 6 (b) shows screen shots on July 3, 2014. It was rainy in western Japan. The icons of umbrella indicates extracted bursty local



areas. If we click or touch these icons, we can observed the details of the selected bursty local areas. An additional movie file shows this in more detail [see Additional file 1].

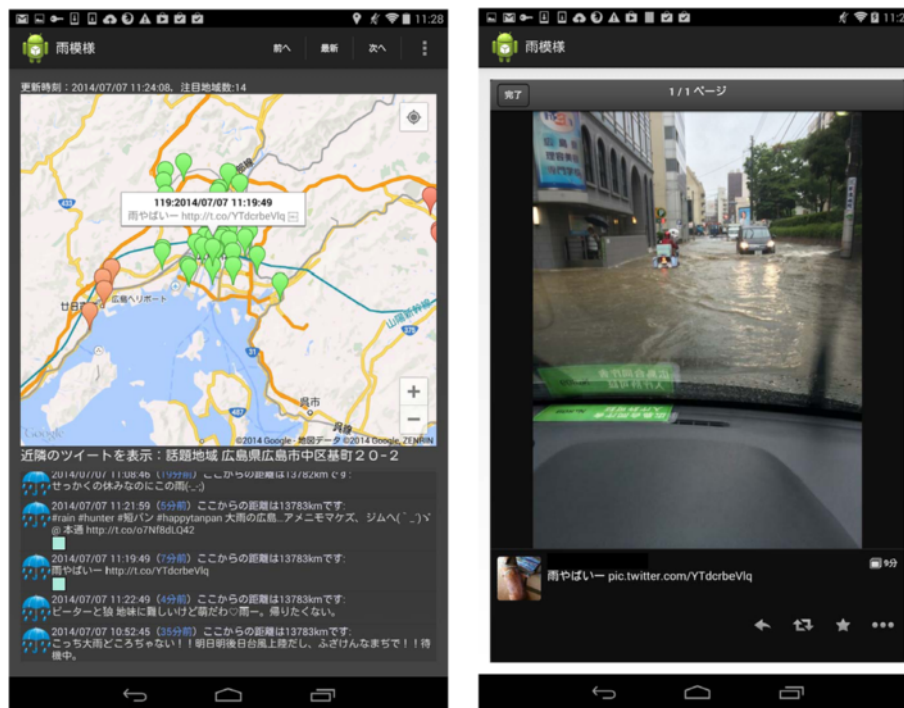
Figure 7 shows the screen shots of the Android application interface to the real-time weather observation system. These screen shots shows the extracted bursty area in Hiroshima on July 7, 2014. On July 7, it rained heavily in Hiroshima. The photo in the screen shot was posted only 5 minutes ago when the screen was captured. In the center of Hiroshima, we observed that flood damage occurred through the posted photo. An additional movie file shows this in more detail [see Additional file 2]. Table 1 shows that Precipitation on July 7, 2014 in Hiroshima. This data is provided by Japan Meteorological Agency. Table 1 indicates that it rained heavily in Hiroshima at 11:00. This is in consistency with this Figure.

**Experimental result**

To evaluate the proposed application, we implemented real-time weather observation system. In our experiments, we used geotagged tweets obtained by crawling

Twitter posts and extracted bursty areas in real time. We collected geotagged tweets from the Twitter site using its Streaming API. In the experiments, we observed two topics “snow” and “rain”. We observed each topic in one of two periods. During January and February 2014, we evaluated whether the proposed application can identify bursty areas according to the topic “snow”. In particular, we focused on 12 days in these two months on which it snowed heavily. During the second period, which comprised June and July 2014, we evaluated whether the proposed application can identify bursty areas according to the topic “rain”. In particular, we focused on 16 days in these two months on which it rained heavily.

In the experiments, we evaluate the performance of the Naive Bayes classifier using cross-validation and the extraction rates of the topics “snow” and “rain” in Japan. Moreover, we observed in real time to clearly the availability of the proposed application. The parameters in the experiment were set as :  $\epsilon = 5 \text{ km}$ ,  $\tau = 3600 \text{ s}$ , and  $MinGDoc$  was 3. We conducted several experiments and we select the best parameters.



**Figure 7** Screen shots of android application interface. This Figure shows the screen shots of the Android application. These screen shots shows the extracted bursty area in Hiroshima on July 7, 2014. The left-side of the figure shows map with extracted clusters. The right-side of the figure shows a photo including in a tweet in an extracted tweets. The tweet says “A heavy rain!”. Moreover, the user who posted this tweet posted the photo that shows a flood on the road. An additional movie file shows this in more detail [see Additional file 2].

**Cross-validation**

Two training data sets *TDS* for “snow” and “rain” were composed of 2,500 geotagged tweets. One consisted of geotagged tweets that included “snow” as a keyword that were posted on February 8. The other consisted of geotagged tweets that included “rain” as a keyword that were posted on June 4. In these two train data sets, the geotagged tweets in the *TDS* were labeled manually. The number of geotagged tweets in the “positive” class, which means including topic “snow”, and the “negative” class was 1648 and 852, respectively. Moreover, the number of geotagged tweets in the “positive” class, which means including the topic “rain”, and the “negative” class was 897 and 1603, respectively.

To evaluate the Naive Bayes classifier, we performed a cross-validation. The number of partitions for the cross-validation was 5, 10, 20, 25, and 50. Figures 8 and 9 show the recall and precision values for each number of partitions for “snow” and “rain”, respectively. The range of recall is from 87% to 89% and from 81% to 84% for “snow”

and “rain”, respectively. The range of precision is from 74% to 76% and from 73% to 74% for “snow” and “rain”, respectively.

On July 3, the Naive Bayes classifier extracted 4738 geotagged tweets that included “rain” as a keyword in the “positive” class. We evaluated this classification. The precision of the 4738 geotagged tweets extracted on July 3 is 93.4%. Therefore, according to the observed geotagged tweets using the Naive Bayes classifier, the proposed framework can extract topic-related geotagged tweets with high precision and recall.

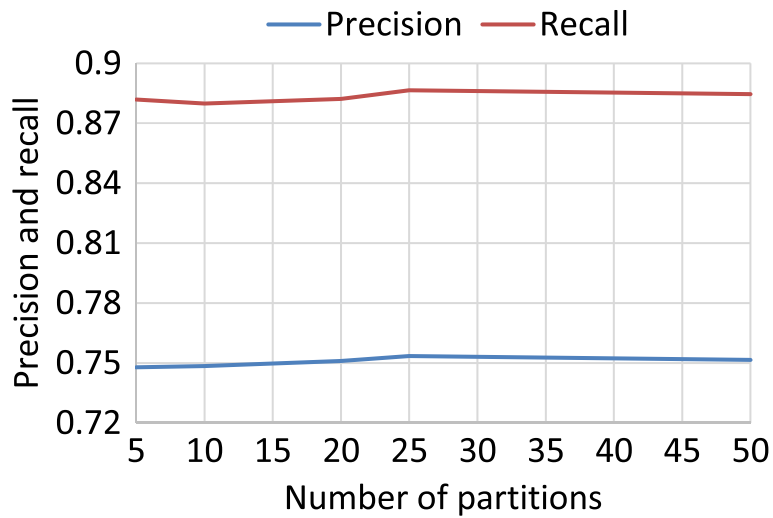
**Extraction rates**

We collected newspaper articles including the topic “snow” and “rain” and extracted areas heavily snowed and rained that were reported from the articles. The number of extracted areas is shown in Tables 2 and 3. There are 106 and 131 areas that are reported to be a heavily rainy and snowy areas in newspaper from January to February and June to July in 2014, respectively. Tables 2 and 3 show the number of crawled geotagged tweets at each day. The average of the number of geotagged tweets is about 300,000 and 350,000.

As mentioned above, the number of heavily snowy and rainy areas were 106 and 131 respectively. Tables 2 and 3

**Table 1 Precipitation on July 7, 2014 in Hiroshima**

Time	6	7	8	9	10	11	12	13	14	15	16	17	18
Precipitation (mm)	0.5	0.5	1.5	20.5	3.0	30.5	10.5	7.5	6.0	3.5	2.5	0.0	0.0



**Figure 8** Cross-validation of “snow”.

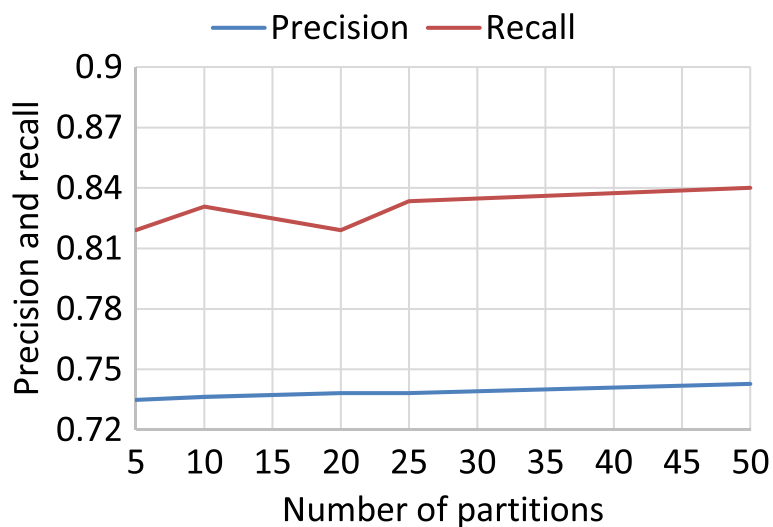
show number of extracted tweets by using the Naive Bayes classifier, number of heavily snowy and rainy areas ( $A$ ), and number of detected heavily snowy and rainy areas ( $B$ ), sorted by date. The value of  $(B/A \times 100)$  indicates the detection rates of the identification. The detection rates are less than 50% for 8 out of 12 days and 8 out of 16 days respectively, because there are many country areas that had heavily where no geotagged tweets were posted.

We also evaluated another type of detection rates; Tables 4 and 5 show the detection rates of detected heavily snowy and rainy areas after removing areas with no posted geotagged tweets. Tables 4 and 5 show  $(B/C \times 100)$  that indicate another detection rates. The detection rates are larger than 50% for 8 out of 12 days and 12 out of 16 days

respectively. The average of detection rates are 83.7% and 73.0%. Therefore, areas where there are geotagged tweets, the proposed framework can identify heavily rainy areas with high detection rates. However, some areas are low detection rates because time interval posted geotagged tweets of those areas are long. Those areas was raining all day on the observed day, for example “Saeki Oita”, “Nobeoka Miyazaki” and “Tsuno Miyazaki” on June 4.

**Real time extraction**

Figure 10 shows that alteration of extracted bursty areas associated with topic “snow” from moment to moment on December 20, 2013. The western part of Japan had first snow in the morning. We observed the system in real time.



**Figure 9** Cross-validation of “rain”.

**Table 2 Data set and detected bursty areas related to topic “snow”**

Date	Number of geotagged tweets	Number of extracted tweets	Number of heavily rainy areas (A)	Number of detected heavily rainy areas (B)	Detection rates (B/A × 100)
2014/1/10	282370	2665	6	3	50.0
2014/1/14	284215	981	1	0	0.00
2014/1/17	283809	995	4	0	0.00
2014/2/6	284065	2821	2	1	50.0
2014/2/8	350867	27823	38	31	81.6
2014/2/10	304380	3424	1	0	0.00
2014/2/11	289628	3564	1	1	100
2014/2/13	306106	3953	1	1	100
2014/2/14	378368	21834	23	17	73.9
2014/2/15	256378	10060	21	15	71.4
2014/2/16	307708	5121	7	2	28.6
2014/2/18	262145	2325	1	1	100

As the expanding snowfall areas, the number of extracted bursty areas increased. We could analyzed and identify which areas had heavy snowfall and what were tweeting. Figure 11 shows that alteration of extracted bursty areas associated with topic “rain” from moment to moment on July 3, 2014. The western part of Japan had heavy storm in the morning; especially, in the northern part of Kyushu,

**Table 3 Data set and detected bursty areas related to topic “rain”**

Date	Number of geotagged tweets	Number of extracted tweets	Number of heavily rainy areas (A)	Number of detected heavily rainy areas (B)	Detection rates (B/A × 100)
2014/6/4	325095	2249	14	3	21.4
2014/6/6	312145	6401	19	7	70.0
2014/6/7	330540	4433	6	2	33.3
2014/6/13	346507	2589	4	0	0.00
2014/6/16	340675	750	4	2	50.0
2014/6/22	411863	4172	2	0	0.00
2014/6/23	355384	700	5	3	60.0
2014/6/25	393441	2331	21	11	52.4
2014/6/29	441959	4838	8	5	62.5
2014/7/3	341770	4738	16	6	37.5
2014/7/7	376734	4173	4	3	75.0
2014/7/8	366887	1405	5	4	80.0
2014/7/9	374707	4704	5	2	40.0
2014/7/10	395061	4803	3	0	0.00
2014/7/11	383704	1763	9	3	33.3
2014/7/19	412403	5369	6	3	50.0

**Table 4 Detected bursty areas related to topic “snow” after removing areas with no posted geotagged tweets**

Date	Number of heavily rainy areas (C)	Number of rainy detected heavily areas (B)	Detection rates (B/C × 100)
2014/1/10	3	3	100
2014/1/14	1	0	0.00
2014/1/17	1	0	0.00
2014/2/6	2	1	50.0
2014/2/8	33	31	93.9
2014/2/10	1	0	0.00
2014/2/11	1	1	100
2014/2/13	1	1	100
2014/2/14	20	17	85.0
2014/2/15	19	15	78.9
2014/2/16	5	2	40.0
2014/2/18	1	1	100

which is located at the west end of Japan, torrential rainfall occurred. Figure 11 shows that many bursty areas were extracted in the northern part of Kyushu at each time.

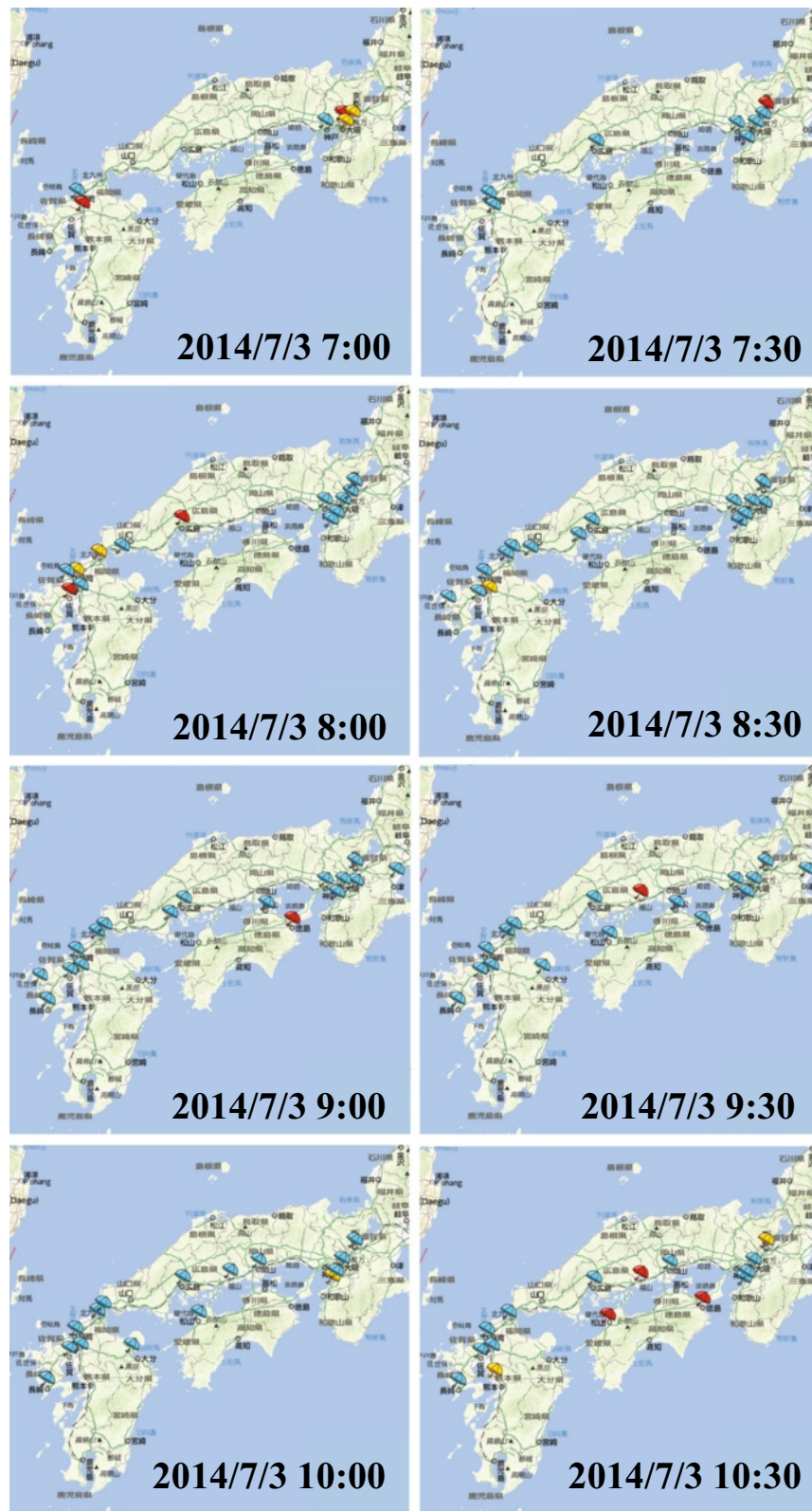
To discuss potential extension of the proposed application, we observed local heavy rain called “Guerrilla Rainstorm”. through the system. Figure 12 shows that a bursty local area extracted in Nagoya City, which is

**Table 5 Detected bursty areas related to topic “rain” after removing areas with no posted geotagged tweets**

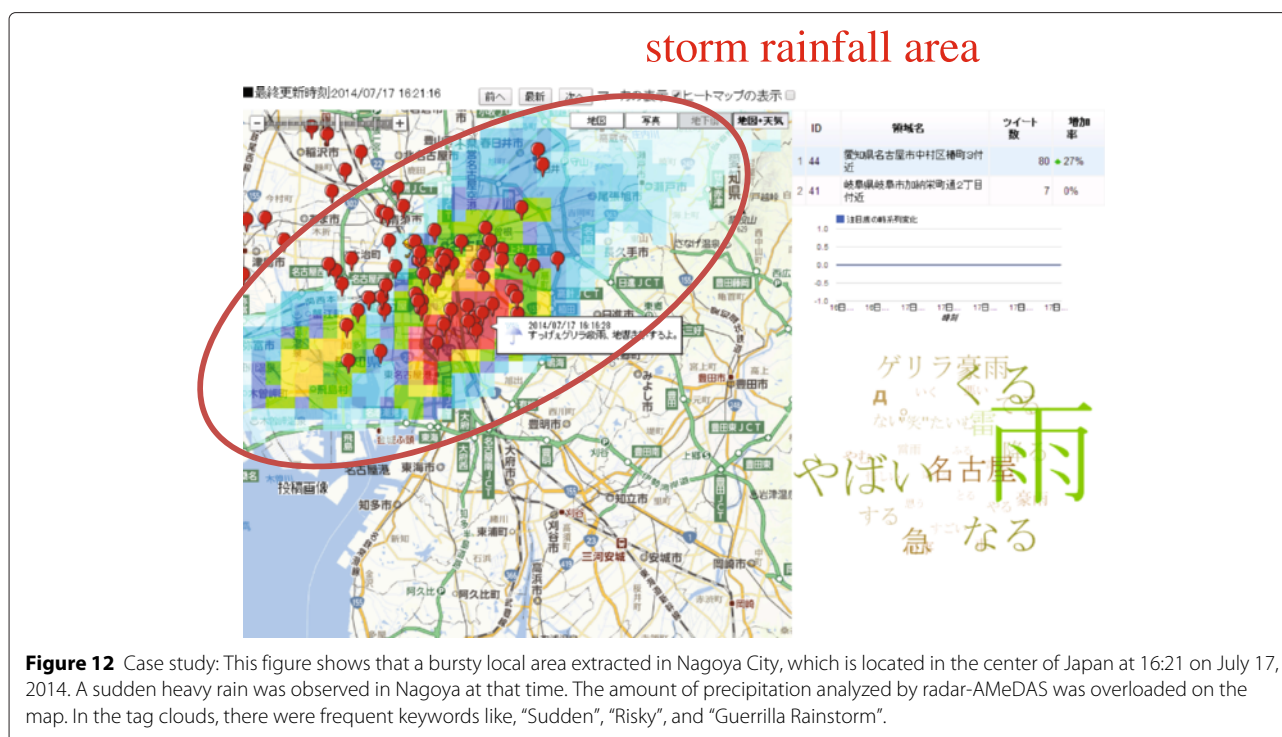
Date	Number of heavily rainy areas (C)	Number of detected heavily rainy areas (B)	Detection rates (B/C × 100)
2014/6/4	8	3	37.5
2014/6/6	9	7	77.8
2014/6/7	3	2	66.7
2014/6/13	1	0	0.00
2014/6/16	3	2	66.7
2014/6/22	1	0	0.00
2014/6/23	3	3	100
2014/6/25	11	11	100
2014/6/29	6	5	83.3
2014/7/3	11	6	54.5
2014/7/7	4	3	75.0
2014/7/8	4	4	100
2014/7/9	3	2	66.7
2014/7/10	0	0	-
2014/7/11	3	3	100
2014/7/19	4	3	75.0



Figure 10 Extracted bursty areas in western Japan on December 20, 2013.



**Figure 11** Extracted bursty areas in western Japan on July 3, 2014.



**Figure 12** Case study: This figure shows that a bursty local area extracted in Nagoya City, which is located in the center of Japan at 16:21 on July 17, 2014. A sudden heavy rain was observed in Nagoya at that time. The amount of precipitation analyzed by radar-AMeDAS was overloaded on the map. In the tag clouds, there were frequent keywords like, “Sudden”, “Risky”, and “Guerrilla Rainstorm”.

located in the center of Japan at 16:21 on July 17, 2014. A sudden heavy rain was observed in Nagoya at that time. The amount of precipitation analyzed by radar-AMeDAS was overloaded on the map. The system could identify a rainfall area as a local bursty area. In the tag clouds, there were frequent keywords like, “Sudden”, “Risky”, and “Guerrilla Rainstorm” appeared in this area. The tag clouds showed that this area was dangerous because of heavy rainfall.

## Conclusion

In this paper, we proposed a novel real-time analysis application for identifying bursty local areas related emergency topics. The aim of our new application is to provide a new platform that can identify and analyze the localities of emergency topics. Three core computational intelligence techniques are applied in our applications: the Naive Bayes classifier technique, the spatiotemporal clustering technique, and the burst detection technique. Moreover, we developed two types of application interface: a Web-based interface and an android application interface. We have implemented a real-time weather observation system embedded the proposed application framework. To evaluate the implemented system, which is embedded the proposed application, actual crawling geotagged tweets posted on the Twitter site were used. We observed real time weather topics and our system could successfully detect bursty areas of observed emergency topics that is related to weather topics. In the future work, we are

planning to develop an alert system that provides the details of detected bursty areas for users located near the bursty areas. We will also extend our proposed method to support multi-languages. Moreover, to improve the usability of our system, we have to develop an automatic method for making train set because making train sets is difficult for end users.

## Additional files

**Additional file 1:** The screen chapter of the weather observation system embedded with the proposed application.

**Additional file 2:** The screen chapter of Figure 7.

## Competing interests

The authors declare that they have no competing interests.

## Authors' contributions

TS and KT proposed the system and implemented the application server. KT implemented the Web application interface and the Android application interface. TS conducted the experiments for evaluation. Both authors read and approved the final manuscript.

## Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 26330139 and Hiroshima City University Grant for Special Academic Research (General Studies).

Received: 31 August 2014 Accepted: 14 January 2015

Published online: 03 April 2015

## References

Abdelhaq H, Sengstock C, Gertz M (2013) Eventweet: Online localized event detection from twitter. *Proc VLDB Endow* 6(12):1326–1329



- Aramaki E, Maskawa S, Morita M (2011) Twitter catches the flu: Detecting influenza epidemics using twitter. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, USA. pp 1568–1576
- Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Second International Conference on Knowledge Discovery and Data Mining. AAI, Palo Alto, CA, USA. pp 226–231
- Hiruta S, Yonezawa T, Jurmu M, Tokuda H (2012) Detection, classification and visualization of place-triggered geotagged tweets. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing UbiComp. '12. ACM, New York, NY, USA. pp 956–963
- Hong L, Ahmed A, Gurumurthy S, Smola AJ, Tsioutsoulis K (2012) Discovering geographical topics in the twitter stream. In: Proceedings of the 21st International Conference on World Wide Web. WWW '12. ACM, New York, NY, USA. pp 769–778
- Hwang M-H, Wang S, Cao G, Padmanabhan A, Zhang Z (2013) Spatiotemporal transformation of social media geostreams: a case study of twitter for flu risk analysis. In: Proceedings of the 4th ACM SIGSPATIAL International Workshop on GeoStreaming. IWGS '13. ACM, New York, NY, USA. pp 12–21
- Jansen BJ, Zhang M, Sobel K, Chowdury A (2009) Twitter power: Tweets as electronic word of mouth. *J Am Soc Inf Sci Technol* 60(11):2169–2188
- Java A, Song X, Finin T, Tseng B (2007) Why we twitter: Understanding microblogging usage and communities. In: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis. WebKDD/SNA-KDD '07. ACM, New York, NY, USA. pp 56–65
- Kaneko T, Yanai K (2013) Visual event mining from geo-tweet photos. In: Multimedia and Expo Workshops (ICMEW) 2013 IEEE International Conference On. IEEE, San Jose, CA, USA. pp 1–6
- Karimi S, Yin J, Paris C (2013) Classifying microblogs for disasters. In: Proceedings of the 18th Australasian Document Computing Symposium. ADCS '13. ACM, New York, NY, USA. pp 26–33
- Kavanaugh A, Fox EA, Sheetz S, Yang S, Li LT, Whalen T, Shoemaker D, Natsev P, Xie L (2011) Social media use by government: From the routine to the critical. In: Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times. dg.o '11. ACM, New York, NY, USA. pp 121–130
- Kleinberg J (2002) Bursty and hierarchical structure in streams. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '02. ACM, New York, NY, USA. pp 91–101
- Mandel B, Culotta A, Boulahanis J, Stark D, Lewis B, Rodrigue J (2012) A demographic analysis of online sentiment during hurricane irene. In: Proceedings of the Second Workshop on Language in Social Media. LSM '12. Association for Computational Linguistics, Stroudsburg, PA, USA. pp 27–36
- Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, New York, NY, USA
- Marcus A, Bernstein MS, Badar O, Karger DR, Madden S, Miller RC (2011) Twitinfo: Aggregating and visualizing microblogs for event exploration. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '11. ACM, New York, NY, USA. pp 227–236
- Miyabe M, Miura A, Aramaki E (2012) Use trend analysis of twitter after the great east japan earthquake. In: Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work Companion. CSCW '12. ACM, New York, NY, USA. pp 175–178
- Murakami A, Nasukawa T (2012) Tweeting about the tsunami?: Mining twitter for information on the tohoku earthquake and tsunami. In: Proceedings of the 21st International Conference Companion on World Wide Web. WWW '12 Companion. ACM, New York, NY, USA. pp 709–710
- Musleh M (2014) Spatio-temporal visual analysis for event-specific tweets. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. SIGMOD '14. ACM, New York, NY, USA. pp 1611–1612
- Ozdikis O, Oguztuzun H, Karagoz P (2013) Evidential location estimation for events detected in twitter. In: Proceedings of the 7th Workshop on Geographic Information Retrieval. GIR '13. ACM, New York, NY, USA. pp 9–16
- Sakai T, Okazaki M, Matsuo Y (2010) Earthquake shakes twitter users: Real-time event detection by social sensors. In: Proceedings of the 19th International Conference on World Wide Web. WWW '10. ACM, New York, NY, USA. pp 851–860
- Sander J, Ester M, Kriegel H-P, Xu X (1998) Density-based clustering in spatial databases: The algorithm gbscan and its applications. *Data Mining Knowl Discov* 2(2):169–194
- Tamura K, Ichimura T (2013) Density-based spatiotemporal clustering algorithm for extracting bursty areas from georeferenced documents. In: Proceedings of The 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013. IEEE Computer Society, Los Alamitos, CA, USA. pp 2079–2084
- Tamura K, Kitakami H (2013) Detecting location-based enumerating bursts in georeferenced micro-posts. In: Proceedings of the 2013 Second IIAI International Conference on Advanced Applied Informatics. IIAI-AAI '13. IEEE Computer Society, Los Alamitos, CA, USA. pp 389–394
- Typhoon real time watcher (2014). <http://typhoon.mapping.jp/>
- Vieweg S, Hughes AL, Starbird K, Palen L (2010) Microblogging during two natural hazards events: What twitter may contribute to situational awareness. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '10. ACM, New York, NY, USA. pp 1079–1088
- Yin J, Lampert A, Cameron M, Robinson B, Power R (2012) Using social media to enhance emergency situation awareness. *Intell Syst IEEE* 27(6):52–59

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---