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

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo

Abstract—Twitter has received much attention recently. An important characteristic of Twitter is its real-time nature. We investigate the real-time interaction of events such as earthquakes in Twitter and propose an algorithm to monitor tweets and to detect a target event. To detect a target event, we devise a classifier of tweets based on features such as the keywords in a tweet, the number of words, and their context. Subsequently, we produce a probabilistic spatiotemporal model for the target event that can find the center of the event location. We regard each Twitter user as a *sensor* and apply particle filtering, which are widely used for location estimation. The particle filter works better than other comparable methods for estimating the locations of target events. As an application, we develop an earthquake reporting system for use in Japan. Because of the numerous earthquakes and the large number of Twitter users throughout the country, we can detect an earthquake with high probability (93 percent of earthquakes of Japan Meteorological Agency (JMA) seismic intensity scale 3 or more are detected) merely by monitoring tweets. Our system detects earthquakes promptly and notification is delivered much faster than JMA broadcast announcements.

Index Terms—Twitter, event detection, social sensor, location estimation, earthquake

1 Introduction

Twitter, a popular microblogging service, has received much attention recently. This online social network is used by millions of people around the world to remain socially connected to their friends, family members, and coworkers through their computers and mobile phones [1]. Twitter asks one question, "What's happening?" Answers must be fewer than 140 characters. A status update message, called a *tweet*, is often used as a message to friends and colleagues. A user can follow other users; that user's followers can read her tweets on a regular basis. A user who is being followed by another user need not necessarily reciprocate by following them back, which renders the links of the network as directed. Since its launch on July 2006, Twitter users have increased rapidly. The number of registered Twitter users exceeded 100 million in April 2010. The service is still adding about 300,000 users per day.[1] Currently, 190 million users use Twitter per month, generating 65 million tweets per day.[2]

Many researchers have published their studies of Twitter to date, especially during the past year. Most studies can be classified into one of three groups: first, some researchers have sought to analyze the network structure of Twitter [2], [3], [4]. Second, some researchers have specifically examined characteristics of Twitter as a social medium [5], [6] . Third, some researchers and developers have tried to create new applications using Twitter [7], [8].

Twitter is categorized as a microblogging service. Microblogging is a form of blogging that enables users to send brief text updates or micromedia such as photographs or audio clips. Microblogging services other than Twitter include Tumblr, Plurk, Jaiku, identi.ca, and others.[3] Our study, which is based on the real-time nature of one social networking service, is applicable to other microblogging services, but we specifically examine Twitter in this study because of its popularity and data volume.

An important characteristic that is common among microblogging services is their real-time nature. Although blog users typically update their blogs once every several days, Twitter users write tweets several times in a single day. Users can know how other users are doing and often what they are thinking about *now*, users repeatedly return to the site and check to see what other people are doing. Several important instances exemplify their real-time nature: in the case of an extremely strong earthquake in Haiti, many pictures were transmitted through Twitter. People were thereby able to know the circumstances of damage in Haiti immediately. In another instance, when an airplane crash-landed on the Hudson River in New York, the first reports were published through Twitter and tumblr.

In such a manner, numerous update results in numerous reports related to *events*. They include social events such as parties, baseball games, and presidential campaigns. They also include disastrous events such as storms, fires, traffic jams, riots, heavy rainfall, and earthquakes. Actually, Twitter is used for various real-time notification such as

1. http://techcrunch.com/2010/06/08/twitter-190-million-users/.
2. http://mashable.com/2010/04/14/twitter-registered-users/.
3. www.tumblr.com, www.plurk.com, www.jaiku.com, identi.ca.

• *The authors are with The University of Tokyo, 2-11-16 Eng. 9, Room 204, Yayoi Bunkyo-ku, Tokyo, Japan.*
E-mail: {sakaki, matsuo}@weblab.t.u-tokyo.ac.jp, okazaki117@gmail.com.

Manuscript received 1 Mar. 2011; revised 16 Nov. 2011; accepted 13 Dec. 2011; published online 13 Feb. 2012.
Recommended for acceptance by A. Tung.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2011-03-0099.
Digital Object Identifier no. 10.1109/TKDE.2012.29.

that necessary for help during a large-scale fire emergency or live traffic updates.

Adam Ostrow, the Editor in Chief at Mashable, a social media news blog, wrote in his blog about the interesting phenomenon of real-time media[4]:

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

This post well represents the motivation of our study. The research question of our study is, "can we detect such event occurrence in real-time by monitoring tweets?"

This paper presents an investigation of the real-time nature of Twitter that is designed to ascertain whether we can extract valid information from it. We propose an event notification system that monitors tweets and delivers notification promptly using knowledge from the investigation. In this research, we take three steps: first, we crawl numerous tweets related to target events; second, we propose probabilistic models to extract events from those tweets and estimate locations of events; finally, we developed an earthquake reporting system that extracts earthquakes from Twitter and sends a message to registered users. Here, we explain our methods using an earthquake as a target event.

First, to obtain tweets on the target event precisely, we apply semantic analysis of a tweet. For example, users might make tweets such as "Earthquake!" or "Now it is shaking," for which *earthquake* or *shaking* could be keywords, but users might also make tweets such as "I am attending an Earthquake Conference," or "Someone is shaking hands with my boss." We prepare the training data and devise a classifier using a Support Vector Machine (SVM) based on features such as keywords in a tweet, the number of words, and the context of target-event words.

After doing so, we obtain a probabilistic spatiotemporal model of an event. We then make a crucial assumption: each Twitter user is regarded as a *sensor* and each tweet as *sensory information*. These virtual sensors, which we designate as *social sensors*, are of a huge variety and have various characteristics: some sensors are very active; others are not. A sensor might be inoperable or malfunctioning sometimes, as when a user is sleeping, or busy doing something else. Consequently, social sensors are very noisy compared to ordinary physical sensors. Regarding each Twitter user as a sensor, the event-detection problem can be reduced to one of object detection and location estimation in a ubiquitous/pervasive computing environment in which we have numerous location sensors: a user has a mobile device or an active badge in an environment where sensors are placed. Through infrared communication or a WiFi signal, the user location is estimated as providing location-based services such as navigation and museum guides [9], [10]. We apply particle filters, which are widely used for location estimation in ubiquitous/pervasive computing [11].

As an application, we develop an earthquake reporting system using Japanese tweets. Japan has numerous earthquakes. Twitter users are similarly numerous and geographically dispersed throughout the country. Therefore, it is sometimes possible to detect an earthquake by monitoring tweets. Our system detects an earthquake occurrence and sends an e-mail, possibly before an earthquake actually arrives at a certain location: An earthquake propagates at about 3-7 km/s. For that reason, a person who is 100 km distant from an earthquake is able to communicate and act for about 20 s before the arrival of an earthquake wave. Moreover, strong earthquakes often cause *tsunami*, which engender more catastrophic disasters than the earthquakes themselves in distant and near places in relation to the earthquake epicenter, as did the Haiti earthquake in 2010 and the Great Eastern Japan earthquake in 2011. Therefore, prompt notification of earthquake occurrences is extremely important to decrease damage by tsunami. In many cases, it could provide notification of tens of minutes or even hours before a tsunami strikes a coastal area.

The contributions of this paper are summarized as follows:

- The paper provides an example of integration of semantic analysis and real-time nature of Twitter, and presents potential uses for Twitter data.
- For earthquake prediction and early warning, many studies have been made in the seismology field. This paper presents an innovative social approach that has not been reported before in the literature.

This paper is organized as described below. In the next section, we explain an investigation of Twitter users and earthquakes in the real world. Section 3 presents our explanation of semantic analysis and sensory information with subsequent the spatiotemporal model in Section 4. In Section 5, we describe the experiments and evaluation of event detection. The earthquake reporting system is introduced in Section 6. Section 7 is devoted to an explanation of related works and discussion. Finally, we conclude the paper.

This paper extends the conference version and includes some elements from it [12].

2 Investigation

We choose earthquakes in Japan as target events, based on the preliminary investigations. We explain them in this section.

First, we choose earthquakes as target events for the following reasons:

1. seismic observations are conducted worldwide, which facilitates acquisition of earthquake information, which also makes it easy to validate the accuracy of our event detection methodology; and
2. it is quite meaningful and valuable to detect earthquakes in earthquake-prone regions.

Second, we choose Japan as the target area based on the following investigation.

4. http://mashable.com/2009/08/12/japan-earthquake/.

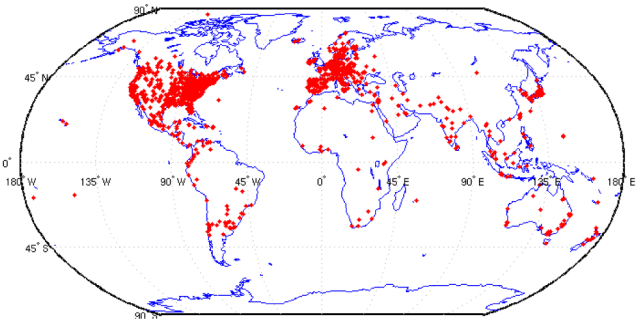


Fig. 1. Twitter user map.

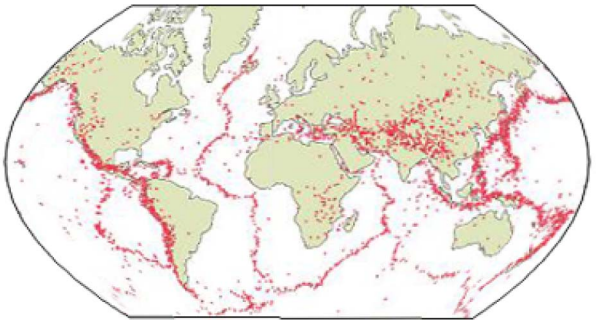


Fig. 2. Earthquake map.

Fig. 1 portrays a map of Twitter users worldwide (obtained from UMBC eBiquity Research Group); Fig. 2 depicts a map of earthquake occurrences worldwide (using data from Japan Meteorological Agency (JMA)). It is apparent that the only intersection of the two maps, those regions with many earthquakes and large Twitter users, is Japan. Other regions such as Indonesia, Turkey, Iran, Italy, and Pacific coastal US cities such as Los Angeles and San Francisco also roughly intersect, but their respective densities are much lower than that in Japan. Many earthquake events occur in Japan and many Twitter users observe earthquakes in Japan, which means that *social sensors* are distributed throughout the country.

We present a brief overview of Twitter in Japan: the Japanese version of Twitter was launched on April 2008. In February 2008, Japan was the No. 2 country with respect to Twitter traffic.[5] At the time of this writing, Japan has the second largest number of tweets (18 percent of all tweets are posted from Japan) in the world.

Therefore, we choose earthquakes in Japan as a target event because of the high density of Twitter users and earthquakes in Japan.

3 Event Detection

As described in this paper, we target event detection. An *event* is an arbitrary classification of a space-time region. An event might have actively participating agents, passive factors, products, and a location in space/time [13]. We target events such as earthquakes, typhoons, and traffic jams, which are readily apparent upon examination of tweets. These events have several properties.

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

Such events include social events such as large parties, sports events, exhibitions, accidents, and political campaigns. They also include natural events such as storms, heavy rains, tornadoes, typhoons/hurricanes/cyclones, and earthquakes. We designate an event we would like to detect using Twitter as a *target event*.

5. http://blog.twitter.com/2008/02/twitter-web-traffic-around-world.html.

In this section, we explain how to detect a target event using Twitter. First, we crawl tweets including keywords related to a target event. From them, we extract tweets that certainly refer to a target event using devices that have been trained with machine learning. Second, we detect a target event and estimate the location from those tweets by treating Twitter users as "social sensors."

3.1 Semantic Analysis of Tweets

To detect a target event from Twitter, we search from Twitter and find useful tweets. Our method of acquiring useful tweets for target event detection is portrayed in Fig. 3.

Tweets might include mention of the target event. For example, users might make tweets such as "Earthquake!" or "Now it is shaking." Consequently, *earthquake* or *shaking* might be keywords (which we call *query words*). However, users might also make tweets such as "I am attending an Earthquake Conference." or "Someone is shaking hands with my boss." Moreover, even if a tweet is referring to the target event, it might not be appropriate as an event report. For instance, a user makes tweets such as "The earthquake yesterday was scary." or "Three earthquakes in four days. Japan scares me." These tweets are truly descriptions of the target event, but they are not real-time reports of the events. Therefore, it is necessary to clarify that a tweet is truly referring to an actual contemporaneous earthquake occurrence, which is denoted as a positive class.

To classify a tweet as a positive class or a negative class, we use a support vector machine [14], which is a widely used machine-learning algorithm. By preparing positive and negative examples as a training set, we can produce a model to classify tweets automatically into positive and negative categories.

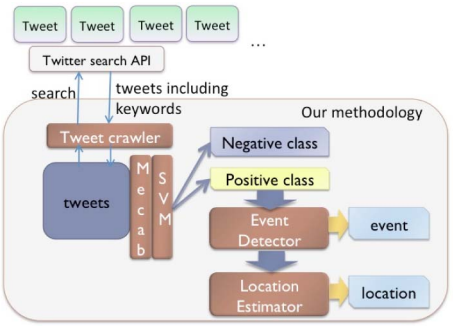


Fig. 3. Method to acquire tweets referred to a target event precisely.

TABLE 1
SVM Features of an Example Sentence

Feature Name	Features
Features A	7 words, the fifth word
Features B	I, am, in, Japan, earthquake, right, now
Features C	Japan, right

We prepare three groups of features for each tweet as described below.

- Features A (statistical features): the number of words in a tweet message, and the position of the query word within a tweet.
- Features B (keyword features): the words in a tweet.[6]
- Features C (word context features): the words before and after the query word.

We can give an illustrative example of these features using the following sentence.

"I am in Japan, earthquake right now!"

(keyword: earthquake)

For this example, Features A, B, C are presented in Table 1.

To process Japanese texts, morphological analysis is conducted using Mecab,[7] which separates sentences into a set of words. For English, we apply standard stop-word elimination and stemming. We compare the usefulness of the features in the discussion in Section 5. Using the obtained model, we can classify whether a new tweet corresponds to a positive class or a negative class.

3.2 Tweet as a Sensory Value

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

Assumption 3.1. *Each Twitter user is regarded as a sensor. A sensor detects a target event and makes a report probabilistically.*

Fig. 4 presents an illustration of the correspondence between sensory data detection and tweet processing. The motivations are the same for both cases: to detect a target event. Observation by sensors corresponds to an observation by Twitter users. They are converted into values using a classifier.

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

6. A tweet is usually short. Therefore, we use every word in a tweet by converting it into a word ID.
7. http://mecab.sourceforge.net/.

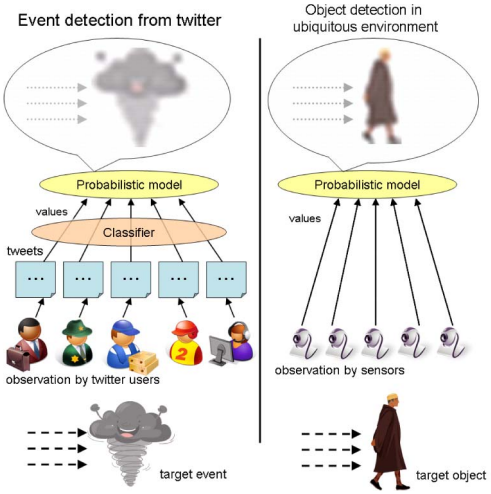


Fig. 4. Correspondence between event detection from Twitter and object detection in a ubiquitous environment.

A tweet can be associated with a time and location: each tweet has its post time, which is obtainable using a search API. In fact, GPS data are attached to a tweet sometimes, such as when a user is using an iPhone. Alternatively, each Twitter user makes a registration on their location in the user profile. The registered location might not be the current location of a tweet. However, we infer it that a person is probably near the registered location. Some tweets include place names in those bodies. Some researchers describe their efforts to extract place names from tweets as a part of Named Entity Recognition [15], [16]. However, the performance derived from those efforts remains insufficient for practical use (precision ranges from 0.6 to 0.8). For the present study, we use GPS data and the registered location of a user. We do not use tweets for spatial analysis if a location is not available; however, we use the tweet information for temporal analyses.

Assumption 3.2. *Each tweet is associated with a time and location, which is a set of latitude and longitude coordinates.*

By regarding a tweet as a sensory value associated with location information, the event detection problem is reduced to detection of an object and its location based on sensor readings. Estimating an object's location is arguably the most fundamental sensing task in many ubiquitous and pervasive computing scenarios [11]. In this research field, some probabilistic models are proposed to detect events and estimate locations by dealing appropriately with sensor readings. The next section explain how these probabilistic models are suited to our tasks of event detection and location estimation.

4 Model

For event detection and location estimation, we use probabilistic models. In this section, we first describe event detection from time-series data. Then we describe the location estimation of a target event.

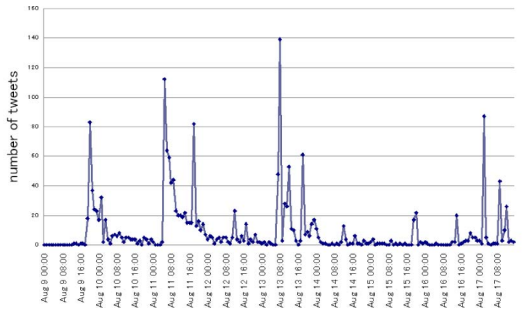


Fig. 5. Number of tweets related to earthquakes.

4.1 Temporal Model

Each tweet has its own post time. When a target event occurs, how do the sensors detect the event? We describe the temporal model of event detection.

First, we examine the actual data. Fig. 5 presents the respective quantities of tweets for a target event: an earthquake. It is apparent that spikes occur in the number of tweets. Each corresponds to an event occurrence. Specifically regarding an earthquake, more than 10 earthquakes occurred during the period.

The distribution is apparently an exponential distribution. The probability density function of the exponential distribution is $f(t;\lambda) = \lambda e^{-\lambda t}$ where $t > 0$ and $\lambda > 0$. The exponential distribution occurs naturally when describing the lengths of the interarrival times in a homogeneous Poisson process.

In the Twitter case, we can infer that if a user detects an event at time 0, then we can assume that the probability of his posting a tweet from t to Δt is fixed as λ. Then, the time to produce a tweet can be regarded as having an exponential distribution. Therefore, even if a user detects an event, she might not make a tweet immediately if she is not online or if she is doing something else. She might make a post only after such problems are resolved. Therefore, it is reasonable that the distribution of the number of tweets follows an exponential distribution. Actually, the data fit an exponential distribution very well. We get $\lambda = 0.34$ on average,

To assess an alarm, we must calculate the reliability of multiple sensor values. For example, a user might produce a false alarm by writing a tweet. It is also possible that the classifier misclassifies a tweet into a positive class. We can design the alarm probabilistically using the following two facts.

- The false-positive ratio p_f of a sensor is approximately 0.35, as we demonstrate in Section 5.1.
- Sensors are assumed to be independent and identically distributed (i.i.d.), as we explain in Section 4.3.

Assuming that we have n sensors, which produce positive signals, the probability of all n sensors returning a false alarm is p_f^n. Therefore, the probability of event occurrence can be estimated as $1 - p_f^n$. Given n_0 sensors at time 0 and $n_0 e^{-\lambda t}$ sensors at time t. Therefore, the number of sensors we expect at time t is

$$\sum_{t_k=0}^{t} n_0 e^{-\lambda t_k} = n_0(1 - e^{-\lambda(t+1)})/(1 - e^{-\lambda}).$$

Consequently, the probability of an event occurrence at time t is

$$p_{occur}(t) = 1 - p_f^{n_0(1-e^{-\lambda(t+1)})/(1-e^{-\lambda})}. \quad (1)$$

We can calculate the probability of event occurrence if we set $\lambda = 0.34$ and $p_f = 0.35$.

4.2 Spatial Model

Each tweet is associated with a location. We describe a method that can estimate the location of an event from sensor readings. To define the problem of location estimation, we consider the evolution of the state sequence $\{x_t, t \in \mathbf{N}\}$ of a target, given that $x_t = f_t(x_{t-1}, u_t), \quad f_t : \mathcal{R}_t^n \times \mathcal{R}_t^n \rightarrow \mathcal{R}_t^n$ where f_t is a possibly nonlinear function of the state x_{t-1}. Furthermore, u_t is an i.i.d. process noise sequence. The objective of tracking is to estimate x_t recursively from measurements, as $z_t = h_t(x_t, n_t), \quad h_t : \mathcal{R}_t^n \times \mathcal{R}_t^n \rightarrow \mathcal{R}_t^n$ where h_t is a possibly nonlinear function, and where n_t is an i.i.d. measurement noise sequence. From a Bayesian perspective, the tracking problem is to calculate, recursively, some degree of belief in the state x_t at time t, given data z_t up to time t.

Presuming that $p(x_{t-1}|z_{t-1})$ is available, the prediction stage uses the following equation.

$$p(x_t|z_{t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|z_{t-1})dx_{t-1}$$

Here, we use a Markov process of order one. Therefore, we can assume that $p(x_t|x_{t-1}, z_{t-1}) = p(x_t|x_{t-1})$.

In the update stage, Bayes' rule is applied as $p(x_t|z_t) = p(z_t|x_t)p(x_t|z_{t-1})/p(z_t|z_{t-1})$ where the normalizing constant is $p(z_t|z_{t-1}) = \int p(z_t|x_t)p(x_t|z_{t-1})dx_t$.

To solve the problem, several methods of Bayesian filters are proposed such as Kalman filters, multihypothesis tracking, grid-based and topological approaches, and particle filters [11]. For this study, we use particle filters, both of which are widely used in location estimation.

4.2.1 Particle Filters

A particle filter is a probabilistic approximation algorithm implementing a Bayes filter, and a member of the family of sequential Monte Carlo methods. For location estimation, it maintains a probability distribution for the location estimation at time t, designated as the belief $Bel(x_t) = \{x_t^i, w_t^i\}, i = 1 \ldots n$. Each x_t^i is a discrete hypothesis related to the object location. The w_t^i are nonnegative weights, called *importance factors*, which sum to one.

The Sequential Importance Sampling (SIS) algorithm is a Monte Carlo method that forms the basis for particle filters. The SIS algorithm consists of recursive propagation of the weights and support points as each measurement is received sequentially.

The algorithm is presented below.

1. **Generation.** Generate and weight a particle set, which means N discrete hypothesis

$$S_0 = \left(s_0^0, s_0^1, s_0^2, \ldots, s_0^{N-1}\right),$$

and allocate them evenly on the map:

$$particle\ s_0^k = (x_0^k, y_0^k, w_0^k)$$

$x : longitude, y : latitude, w : weight.$

2. **Resampling.** Resample N particles from a particle set S_t using weights of respective particles and allocate them on the map. (We allow resampling of more than that of the same particles.).
3. **Prediction.** Predict the next state of a particle set S_t from Newton's motion equation

$$\begin{aligned}(x_t^k, y_t^k) &= \Big(x_{t-1}^k + v_{x_{t-1}}\Delta t + \frac{a_{x_{t-1}}}{2}\Delta t^2,\\ &\qquad y_{t-1}^k + v_{y_{t-1}}\Delta t + \frac{a_{y_{t-1}}}{2}\Delta t^2\Big)\\ (v_{x_t}, v_{y_t}) &= (v_{x_{t-1}} + a_{x_{t-1}}, v_{y_{t-1}}, a_{y_{t-1}})\\ a_{x_t} &= \mathcal{N}(0;\sigma^2), \quad a_{y_t} = \mathcal{N}(0;\sigma^2).\end{aligned}$$

4. **Weighing.** Recalculate the weight of S_t by measurement $m(m_x, m_y)$ as follows:

$$\begin{aligned}dx_t^k &= m_x - x_t^k, \quad dy_t^k = m_y - y_t^k\\ w_t^k &= \frac{1}{(\sqrt{2\pi}\sigma)}\\ &\quad \cdot exp\left(-\frac{(dx_t^{k2} + dy_t^{k2})}{2\sigma^2}\right).\end{aligned}$$

5. **Measurement.** Calculate the current object location $o(x_t, y_t)$ by the average of $s(x_t, y_t) \in S_t$.
6. **Iteration.** Iterate Steps 2, 3, 4, and 5 until convergence.

4.2.2 Consideration of Sensor Geographic Distribution

We must consider the sensor geographic distribution to treat readings of social sensors more precisely.

In location estimation by physical sensors, those sensors are located evenly in many cases. We can treat sensor readings equally in such situations. Actually, social sensors are not placed evenly in many cases because social media users are concentrated in urban areas. In Japan, most users live in Tokyo. Therefore, we should incorporate the geographic distribution of social sensors into spatial models.

It is thought that there are fewer social sensors in areas where fewer Twitter users live. Consequently, those sensors have lower probabilities to response value. In spite of such low probabilities, if a sensor in a less-populated area produce a positive value to one earthquake, then it can be inferred that the center of the earthquake is close to that sensor. Therefore, we assume that sensor values in less-populated areas are more important than those in densely populated areas. Based on this assumption, we calculate weights of respective particles based on the geographic distribution of social sensors.

We use a more advanced algorithm with resampling [17]. We use the weight distribution $D_w(x, y)$, as obtained from the Twitter user distribution, to examine the biases of user locations.[8] We customize the algorithm related to particle filters as follows:

8. We sample tweets associated with locations and obtain a user distribution that reflects the numbers of tweets in respective regions.

1. We collect Twitter users randomly along with their location information.
   $s_j(x_{s_j}, y_{s_j})(s_j \in S)$: longitude and latitude of $user_j$.
   N_s: Number of users we collect.
2. In the Generation step, we weight each particle based on weight distribution $D_w(x_k, y_k)$ after they are allocated

$$\begin{aligned}(dx_{k,s_j}, dy_{k,s_j}) &= (x_k - x_{s_j}, y_k - y_{s_j})\\ D_w(x_k, y_k) &= \sum_{j=1}^{N_s} \frac{1}{(\sqrt{2\pi}\sigma)}\\ &\quad \cdot exp\left(-\frac{({dx_{k,s_j}}^2 + {dy_{k,s_j}}^2)}{2\sigma^2}\right).\end{aligned}$$

3. In the Weighing step, we calculate the weights of each particle using the following equation:

$$\begin{aligned}w_t^k &= D_w(x_t^k, y_t^k) \cdot \frac{1}{(\sqrt{2\pi}\sigma)}\\ &\quad \cdot exp\left(-\frac{(dx_t^{k2} + dy_t^{k2})}{2\sigma^2}\right).\end{aligned}$$

As described in this paper, we designate this customized method as a *weighted particle filter*.

4.2.3 Techniques to Speed up the Process

As described in this paper, we want to estimate location of events quickly as soon as possible because one objective of this research is to develop a real-time earthquake detection system. Therefore, we must decrease the time complexity of methods used for location estimation.

The time complexity of a normal particle filter is expressed as $O(N_p N_m)$(N_p, number of particles; N_m, number of observations). The time complexity of the weighted particle filter is expressed as $O(N_p N_m N_s)$ (N_s, number of sensors to calculate the geographic distribution). In the pre-examination, we set $N_p = 2{,}000$, $N_m = 20$, $N_s = 6{,}421$. It takes less than 1 s to estimate the location of an earthquake center using a normal particle filter. It takes less from 1 minute to 3 minute to estimate the location of an earthquake center by weighted particle filter. Therefore, we want to decrease N_s to calculate the location of earthquake centers more quickly.

As described in this paper, we sample some users from all users to calculate the sensor geographic distribution and produce a new set of S^* users. We apply the following three approaches.

- **Sampling:** sample N_{s^*} users from S randomly and designate them as S^*.
- **Sampling and average:**
  - Sample m_{sample} users from S randomly and calculate an average position for it ($P_{a_l}(x_l, y_l)$)

$$P_{a_l}(x_l, y_l) = \left(\frac{1}{m_{sample}}\sum_{i=1}^{m_{sample}} x_{l,i}, \frac{1}{m_{sample}}\sum_{i=1}^{m_{sample}} y_{l,i}\right).$$

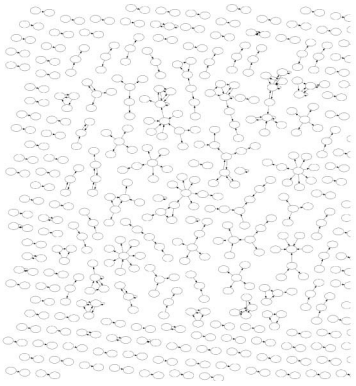


Fig. 6. Earthquake information diffusion network.

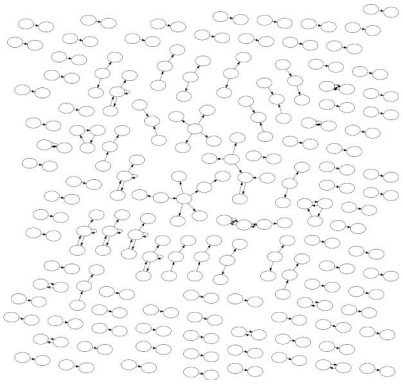


Fig. 7. Typhoon information diffusion network.

- Repeat step 1 N_{s^*} times and designate those points of average positions as $S^* = P_l(l = 0 \ldots N_{s^*})$.

- **Sampling and mean:**
  - Sample m_{sample} users from S randomly and calculate a mean position of it($P_{m_l}(x_l, y_l)$)

$$P_{m_l}(x_l, y_l) = \left(\frac{1}{2}(Max(x_{l,i}) + Min(x_{l,i})) \right. \\ \left. \frac{1}{2}(Max(y_{l,i}) + Min(y_{l,i}))\right).$$

  - Repeat step 1 N_{s^*} times and designate those points of mean positions as S^*.

4.3 Information Diffusion Related to a Real-Time Event

Some information related to an event diffuses through Twitter. For example, if a user detects an earthquake and makes a tweet about the earthquake, then a follower of that user might make tweets about that. This characteristic is important because, in our model, sensors might not be mutually independent, which would have an undesired effect in terms of event detection.

Figs. 6, 7, and 8, respectively, portray the information flow networks for an earthquake, a typhoon, and a new Nintendo DS game.[9]

We infer an information flow between two users: assume that user A follows user B. If user B makes a tweet about an event, and soon thereafter user A makes a tweet about an event, then we consider that the information flows from B to A.[10] This definition is similar to those used in other studies of information diffusion (e.g., [18], [19]).

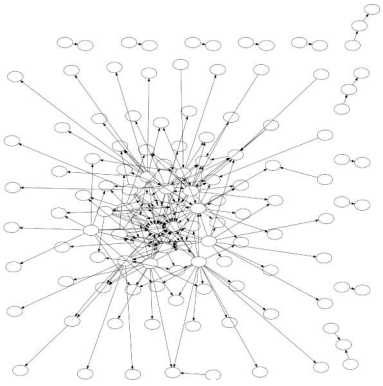


Fig. 8. New Nintendo game information diffusion network.

We define networks of two types.

1. Follower networks: networks express the following relations among users.
   a. node: users posted tweets about target events.
   b. edge: user A follows user B.
2. Information flow networks: networks express information flows among users.
   a. node: users posted tweets about target events.
   b. edge: user A follows user B and user A makes a tweet about an event after user B makes a tweet.

For the cases presented in Figs. 6 and 7, earthquakes and typhoons, very little information diffusion takes place on Twitter. In contrast, Fig. 8, which shows aspects of the release of a new game, reflects the scale and rapidity of information diffusion. We crawl tweets including the name of the game during one week in September 2009. Information about the game propagates among many users. Users are not i.i.d. when they post tweets about topics of such kinds. To verify these facts numerically, we define one index $R_{PageRank}$ as follows:

$$R_{PageRank} = \frac{PageRank_{follower}}{PageRank_{flow}}. \quad (2)$$

PageRank is a measure of network centrality. It is said that information diffusion tends to occur in networks that have nodes with high PageRank[20]. $PageRank_{follower}$ signifies the max PageRank value of follower networks. $PageRank_{flow}$ stands for the max PageRank value of information flow networks. $R_{PageRank}$ represents the difference of PageRank between information flow networks of an event and follower networks of the same event. If $R_{PageRank}$ is high, then an information flow network has no node with a high degree of connectivity, which means that information diffusion does not occur so much in relation to the event in the Twitter world. Fig. 9 shows $R_{PageRank}$ of 20 events, including 15 news events, 3 earthquakes, and

9. Love Plus, a game that offers a virtual girlfriend experience, was released on September 3, 2009.

10. Because of this definition, the diffusion includes *retweet*, which is a type of message that repeats some information that was tweeted previously by another user.

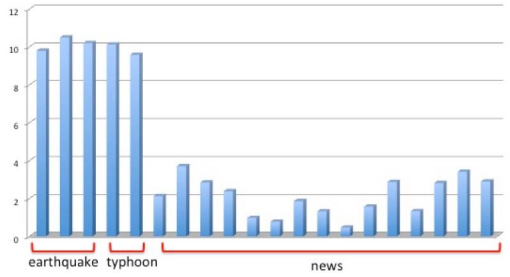


Fig. 9. $R_{PageRank}$ of networks of earthquakes, typhoons, and news events.

2 typhoons. In Fig. 9, typhoons and earthquakes have high $R_{PageRank}$. This demonstrates that information diffusion does not occur on earthquakes and typhoons.

Therefore, we can assume that the sensors are i.i.d. when considering real-time event detection such as typhoons and earthquakes. Additionally, we must verify sensors are i.i.d or not when we apply our proposed method to some events.

5 Experiments and Evaluation

In this section, we describe the experimentally obtained results and evaluation of tweet classification and location estimation.

The whole algorithm is the following:

1. Given a set of queries Q for a target event.
2. Put a query Q using search API every s seconds and obtain tweets T.
3. For each tweet $t \in T$, obtain features A, B, and C. Apply the classification to obtain value $v_t = \{0, 1\}$.
4. If the enough number of tweets comes(p_{occur} in (1) exceeds 0.99 under the condition: *10 tweets in 10 minutes*; $\lambda = 0.34$; $p_f = 0.35$;) then proceed to step 5.
5. For each tweet $t \in T$, we obtain the latitude and the longitude l_t by 1) using the associated GPS location, 2) making a query to Google Map for the registered location for user u_t. Set $l_t = \text{null}$ if neither functions.
6. Calculate the estimated location of the event from $l_t, t \in T$ using normal particle filtering, particle filtering with assigned weights, and particle filtering with weights and sampling.
7. Send alert e-mails to registered users.

We prepare a set of queries Q for a target event. We first search for tweets T including the query set Q from Twitter every s seconds. We use a search API[11] to search tweets. In the earthquake case, we set $Q = \{''\text{earthquake}'' \; and \; ''\text{shaking}''\}$; in the typhoon case, we set $Q = \{''typhoon''\}$. We set s as 3 s. After determining a classification and obtaining a positive example, the system makes a calculation of a temporal and spatial probabilistic model. The location information of each tweet is obtained and used for location estimation of the event. The earthquake reporting system explained in the next section quickly sends an e-mail (usually mobile e-mail) to registered users.

11. search.twitter.com.

TABLE 2
Classification Performance

(i) *earthquake* query:

Features	Recall	Precision	F-value
A	87.50%	63.64%	73.69%
B	87.50%	38.89%	53.85%
C	50.00%	66.67%	57.14%
All	87.50 %	63.64%	73.69%

(ii) *shaking* query:

Features	Recall	Precision	F-value
A	66.67%	68.57%	67.61%
B	86.11%	57.41%	68.89%
C	52.78%	86.36%	68.20%
All	80.56 %	65.91%	72.50%

5.1 Evaluation by Semantic Analysis

For classification of tweets, we prepared 597 positive examples that report earthquake occurrence as a training set (the size of the training set is not large but we think it is enough because our event detection system performs well with satisfactory accuracy as we will describe later). The classification performance is presented in Table 2. We use two query words: *earthquake* and *shaking*. Performance results obtained using respective queries are shown. We used a linear kernel for SVM. We obtain the highest F-value when we use feature A and all features. Surprisingly, features B and C do not contribute much to the classification performance. When an earthquake occurs, a user becomes surprised and might produce a very short tweet. It is apparent that the recall is not as high as the precision. That result is attributable to the usage of query words in a different context than we had intended. Sometimes it is difficult even for humans to judge whether a tweet is reporting an actual earthquake or not. Some examples are that a user might write "Is this an earthquake or a truck passing?" Overall, the classification performance is good considering that we can use multiple sensor readings as evidence for event detection.

5.2 Evaluation of Spatial Estimation

Fig. 10 presents the location estimation of an earthquake that occurred on August 11. Many tweets originated from over a wide region in Japan. The estimated location of the earthquake (shown as estimation by weighed particle filter) is close to the actual epicenter of the earthquake, which shows the efficiency of the location estimation algorithm. Table 3 presents results of location estimation based on a total of 621 tweets for 25 earthquakes that occurred during August-October 2009. We compare results obtained using three particle filtering methods with the weighted average and the median as a baseline. The weighted average simply takes the average of latitudes and longitude on all the positive tweets; median simply takes their median. Particle filters of three kinds perform well compared to other baseline methods. Particle filter with weighting works better than the normal particle filter. The performance of particle filter with weighting and sampling is similar to that of the particle filter with weighting when $N_s = 100$(N_s, number of samples) and sampling by *mean value* method.

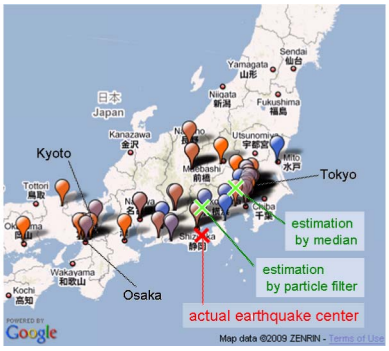


Fig. 10. Earthquake location estimation based on tweets. Balloons show the tweets related to an earthquake. The cross shows the earthquake epicenter. Red represents early tweets; blue shows later tweets.

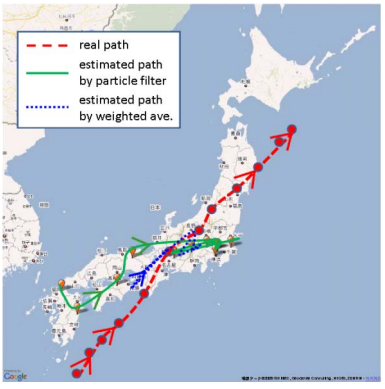


Fig. 11. Typhoon trajectory estimation based on tweets.

In Figs. 12 and 13, data are shown for comparison of the weighted particle filter and the sampled particle filter with each sampling method in performance and time complexity. Mean values work better than other sampling methods do. The performance of the sampled particle filter with mean value has a positive correlation with N_s; it converges $N_s = 300$, which means that 5 percent of all sensors are sufficient for sampling. It takes $6.4\,\mathrm{s}$ for calculation by sampled particle filter with $N_s = 300$; it takes $120\,\mathrm{s}$ for calculation using the weighted particle filter. We can perform computations 20 times faster than before with only a slight drop in performance.

Results show that if the center of the earthquake is in an oceanic area, it is more difficult to locate it precisely from tweets. Similarly, it becomes more difficult to produce good estimations in less-populated areas. That result is reasonable: all other things being equal, the greater the number of sensors, the more precise the estimation will be.

Fig. 11 depicts a trajectory estimation of typhoon Melor based on a total of 2,037 tweets. For an earthquake, the center is one location. However, for a typhoon, the center moves, producing a trajectory. The relative performance of

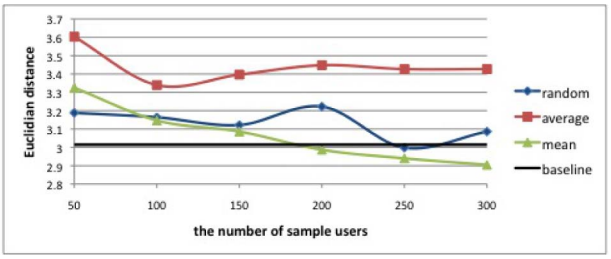


Fig. 12. Performances of the weighted particle filter and the sampled particle filters with respective sampling methods; x-axis: number of samples; y-axis: euclidean distance error.

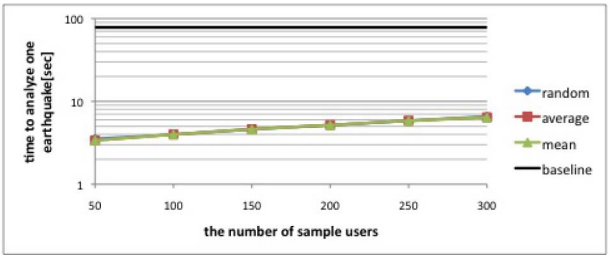


Fig. 13. Time complexity of weighted particle filter and sampled particle filters with each sampling method: x-axis, number of samples; y-axis, mean squared error.

several methods is presented in Table 3. The particle filter works well and outputs a trajectory path resembling the actual path of the typhoon. (Tables in the Supplemental Material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2012.29, present detailed figures of results.)

6 Event Detection System

We developed earthquake detection systems using our methodology of event detection, "Toretter." In this section, we present this system and explain its features.

6.1 Earthquake Reporting System

We developed an earthquake-reporting system using the event detection algorithm. Earthquake information is much more valuable if it is received in real time. Given some amount of advanced warning, any person would be able to turn off a stove or heater at home and then seek protection under a desk or table if such a person were to have several seconds' notice before an earthquake actually strikes an area. It goes without saying that, for such a warning, earlier is better.

Vast amounts of work have been done on intermediate-term earthquake prediction in the seismology field (e.g., [21]). Various attempts have also been undertaken to produce short-term forecasts to realize an earthquake warning system by observing electromagnetic emissions

TABLE 3
Location Estimation Accuracy of Earthquakes and a Typhoon Trajectory from Tweets

	Median	Weighted ave.	Particle (normal)	**Particle (weight)**	Particle (sampled)
earthquakes	5.47	3.62	3.85	**3.01**	**3.14**
typhoon trajectory	4.39	4.02	9.56	**3.58**	**3.67**

For each method, we present the difference of the estimated latitude and the longitude to the actual ones, and their euclidean distance. Smaller distance reflects better performance. With the sampled particle filter, we use mean values for sampling method, and sampled 300 users.

Fig. 14. Screenshot of Toretter, an earthquake reporting system.

from ground-based sensors and satellites [22]. In Japan, the government has allocated a considerable amount of its budget to mitigating earthquake damage. In fact, an earthquake early warning service has been operated by JMA since 2007. It provides advance announcements of the estimated seismic intensities and expected arrival times.

6.2 Proposed System

The proposed system, called *Toretter*,[12] has been operated since August 8, 2010. A system screenshot is depicted in Fig. 14. Users can see the detection of past earthquakes. They can register their e-mails to receive notices of future earthquake detection reports.

It alerts users and urges them to prepare for the imminent earthquake. It is hoped that a user receives the e-mail before the earthquake actually affects that area.

We evaluate various conditions under which alarms might be sent to choose better parameters for our proposed system. We set alarm conditions as N_{tweet} *positive tweets comes in 10 minute*. We evaluate those methods by $Precision = \frac{N_{earthquake}}{N_{alarms}}$ and $Recall = \frac{N_{earthquake}}{All_{earthquake}}$ ($N_{earthquake}$: Number of earthquakes detected correctly, N_{alarms}: number of alarms, $All_{earthquake}$: number of all earthquakes that occurred).

Fig. 15 shows the performance of our system in each alarm condition using 1,136 earthquakes during 19 months from Aug 2009 to Feb 2011. We evaluate our system when we set $N_{tweet} = 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 110$. Judging from results in Fig. 15 , the precision and the recall of our system is trade off. We detected 93 percent of earthquakes that were stronger than JMA seismic intensity scale[13] 3 or higher when we set $N_{tweet} = 10$ (In the middle graph of Fig. 15. However, the precision is very low, which means the system produces many false-positive alarms in such cases. While, if we set $N_{tweet} = 100$, we can detect only 80 percent of earthquakes stronger than scale 3, but 75 percent of alarms are correct.

We investigated the reasons underlying errors of our system. These errors are divided into errors of two types.

12. It means "we have taken it" in Japanese.

13. The JMA seismic intensity scale is a measure used in Japan and Taiwan to indicate earthquake strength at a certain location. Unlike the Richter magnitude scale, the JMA scale reflects the degree of shaking at a point on the earth's surface.

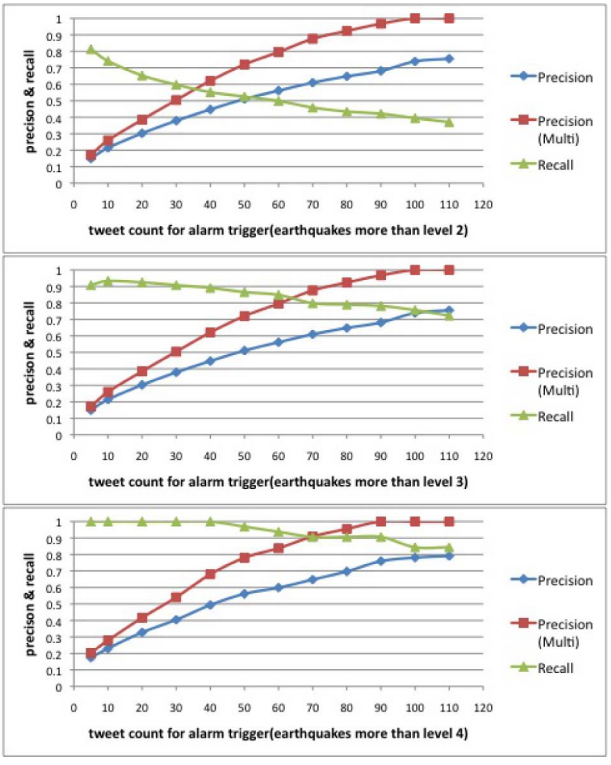


Fig. 15. Earthquake detection performance for 19 months from Aug 2009 to Feb 2011. Top : performance for earthquakes more than scale 2. Middle : performance for earthquakes more than scale 3. Bottom : performance for earthquakes more than scale 4. x-axis: the number of tweets need to make an alarm in 10 minutes.

The first type is the case of detecting one earthquake several times. We designate such errors as "multiple detection." The second type includes cases other than "multiple detection." We designate this type as "incorrect detection." Table 4 shows rates of multiple detection for each JMA seismic intensity scale rating. From Table 4, large earthquakes engender multiple detection. It is thought that people post more tweets for a longer period after strong earthquakes.

We ignore errors by multiple detection and recalculate the precision of our system. (If people receive several alarms in short time span, they can understand that those alarm come from the same earthquake). These results are presented as "Presented(Multi)" in Fig. 15. The precision increase by about 20 percent after we remove an affection of multiple detection errors.

Judging from the objective of this research, our system must detect all strong earthquakes (stronger than scale 4) and produce fewer false-positive alarms. Therefore, we should set $N_{tweet} = 40$ to warn people to escape from a series of events caused by the earthquake.

TABLE 4
Earthquake Detection Performance for Two Months from August 2009

JMA intensity scale	2	3	4 or more
No. of earthquakes	53	22	5
No. of multiple detections	2(3.8%)	7(31.8%)	5(100%)

"Multiple detection errors."

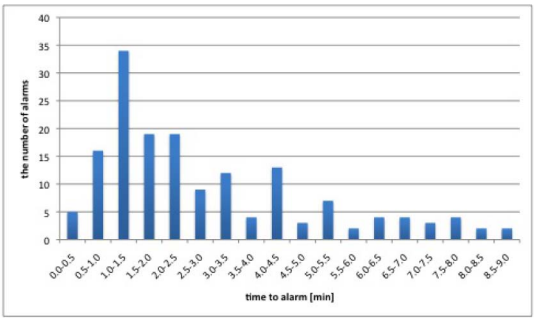


Fig. 16. Frequent distribution of the time to send alarm e-mail. (x-axis: time to send e-mail[sec] y-axis: frequency of earthquakes, alarm trigger: 40 tweets in 10 minutes).

Fig. 16 presents the frequency distribution of the time to send notification of earthquakes by e-mail during one year, all of which our system detected. The first tweet of an earthquake is usually made within a minute or so. The delay can result from the time for posting a tweet by a user, the time to index the post in Twitter servers, and the time to make queries using our system. Our system sent e-mails within a minute and half on average, and sent 13 percent of them within a minute. That delivery is far faster than the rapid broadcasts of announcements of JMA, which are widely broadcast on TV; on average, a JMA announcement is broadcast 6 minutes after an earthquake occurs.

Based on these results, we infer that our system probably has a high recall rate and medium precision. Sometimes the system produces a false alarm when a strong earthquake occurs or when several earthquakes occur during a single day. The current system uses only a static condition for giving an alarm: "N_{tweet} tweets within 10 minutes." We must change the use of this condition dynamically to increase the precision of the system, particularly in terms of the repetition and intensity of earthquakes.

7 Related Work

Twitter is an interesting example of the most recent type of social media. Numerous researchers have examined Twitter.

Regarding similar research to that presented in this paper, some researchers have attempted topic detection using Twitter. Cataldi et al. proposed a novel method to detect emerging topics using a keyword-based topic graph [23]. They succeeded in detecting news keywords that are popular in Twitter. For instance, Eyjafjallajokull (a volcano in Iceland) and Samaranch (the previous President of IOC, who died in April 2010). Marc et al. divided increasingly popular keywords on Twitter into patterns of various kinds using SOM, thereby demonstrating that Twitter users contribute to the discussion of these trends.

Aside from the studies introduced in Section 1 and these studies, several others have been done. We classify studies dealing with Twitter or data on Twitter into three groups.

First, some researchers specifically examine the network structure of Twitter and investigate Twitter network features of various kinds. Java et al. analyzed Twitter as early as 2007. They described the social network of Twitter users and investigated the motivations of Twitter users [2]. Haewoon et al. crawled a vast amount of Twitter data, analyzed the Twitter follower-following topology and ranked users by Pagerank [4]. Huberman et al. analyzed more than 300 thousand users. They discovered that the relation between friends (defined as a person to whom a user has directed posts using an "@" symbol) is the key to understanding interaction in Twitter [3].

Second, some researchers have examined characteristics of Twitter as social media. Recently, Boyd et al. have continued their investigation of *retweet* activity, which is the Twitter-equivalent of e-mail forwarding, by which users post messages that were originally posted by others [5]. Tumasjan et al. crawled many tweets referring to the election in Germany and attempted to predict the results of the election: which political parties would win the election [6]. Óconnor extracts public opinion from Twitter using sentiment analysis and reports the possibility of using a proposed method instead of polls [24].

Third, some studies elucidate the benefits of novel applications of Twitter: Ebner and Schiefner establish a microblogging community and studies how to use Twitter as a tool for mobile e-learning [25]. The integration of the Semantic Web and microblogging was described in a previous report [26] in which a distributed architecture is proposed and the contents are aggregated.

In contrast to the small number of academic studies of Twitter, numerous Twitter applications exist. Some are used for analyses of Twitter data. For example, Tweettronics[14] provides an analysis of tweets related to brands and products for marketing purposes. It can classify positive and negative tweets, and it can identify influential users. The classification of tweets might be done similarly to our algorithm. Web2express Digest[15] is a website that autodiscovers information from Twitter streaming data to find real-time interesting conversations. It also uses natural language processing and sentiment analysis to discover interesting topics, as we do in our study.

Various studies have analyzed web data (aside from that of Twitter), particularly addressing its spatial aspects. The most relevant study to ours is one by Backstrom et al. [27]. That study used queries with location (obtained by IP addresses), and presented a probabilistic framework for quantifying spatial variation. The model is based on a decomposition of the surface of the earth into small grid cells. The framework finds a query's geographic center and spatial dispersion. Although the motivation is very similar to that which spurs our study, the events to be detected differ. Some examples are that people might not make a search query *earthquake* when they experience an earthquake. Therefore, our approach complements their work. Similarly to our work, Mei et al. targeted blogs and analyzed their spatiotemporal patterns [28]. They presented examples for Hurricane Katrina, Hurricane Rita, and the iPod Nano (Apple Computer Inc.). The motivation of that study is similar to ours, but Twitter data are more time sensitive; our study examines even more time-critical events such as earthquakes.

Some studies have specifically investigated collaborative bookmarking data, as Flickr provides, from a spatiotemporal perspective: Serdyukov et al. describes investigations of

14. http://www.tweettronics.com.
15. http://web2express.org.

generic methods for placing photographs from Flickr on the world map [29]. Rattenbury et al. [30] specifically examines the problem of extracting place and event semantics for tags that are assigned to photographs on Flickr. They propose scale-structure identification, which is a burst-detection method based on scaled spatial and temporal segments.

Location estimation studies are often done in the field of ubiquitous computing. Estimating an object's location is arguably the most fundamental sensing task in many ubiquitous and pervasive computing scenarios. Representing locations probabilistically provides a unified interface for location information, which enables us to produce applications that are independent of the sensors used, even when using starkly different sensor types such as GPS and infrared badges [11], or even Twitter. Kalman filters, multihypothesis tracking, grid-based, and topological approaches, and particle filters are well-known algorithms used for location estimation. Hightower and Borriello described the application of particle filters to location sensors deployed throughout a lab building [31]. More than 30 lab residents were tracked. Then their locations were estimated accurately using the particle filter approach.

8 Discussion

Many studies have been undertaken to monitor the social situation by treating participants in social media, such as those using Twitter, as social sensors. However, most such studies are aimed at observation of long-term changes of social situations. Our research is an early approach to use Twitter as a social sensor for detection of real-time events.

Additionally, it is meaningful that we apply methods for event detection using ordinal physical sensors for event detection by social sensors. The field of event detection using physical sensors has already been developed. Methods of many kinds exist in the field. Therefore, it is possible that events of many kinds can be observed from Twitter through application of those methods. Our research has produced one of the first approaches to use such methods.

We intend to expand our system to detect events of various kinds using Twitter.

Our model includes the assumption that a single instance of the target event exists. For example, we assume that plural earthquakes or typhoons do not occur simultaneously. Although that assumption is reasonable for these cases, it might not hold for other events such as traffic jams, accidents, and rainbows. To realize multiple event detection, we must produce advanced probabilistic models that can accommodate multiple event occurrences.

A search query is important for seeking tweets that might be relevant. For example, we set query terms as *earthquake* and *shaking* because most tweets mentioning an earthquake occurrence use either word. However, to improve the recall, it is necessary to obtain a good set of queries. In fact, advanced algorithms can be useful for query expansion, which remains as a subject of our future work.

9 Conclusion

As described in this paper, we investigated the real-time nature of Twitter, devoting particular attention to event detection. Semantic analyses were applied to tweets to classify them into a positive and a negative class. We regard each Twitter user as a sensor, and set the problem as detection of an event based on sensory observations. Location estimation methods such as particle filtering are used to estimate the locations of events. As an application, we developed an earthquake reporting system, which is a novel approach to notify people promptly of an earthquake event.

Microblogging has real-time characteristics that distinguish it from other social media such as blogs and collaborative bookmarks. As described in this paper, we presented an example that leverages the real-time nature of Twitter to make it useful in solving an important social problem: natural disasters. It is hoped that this paper will provide some insight into the future integration of semantic analysis with microblogging data.

References

[1] M. Sarah, C. Abdur, H. Gregor, L. Ben, and M. Roger, "Twitter and the Micro-Messaging Revolution," technical report, O'Reilly Radar, 2008.

[2] A. Java, X. Song, T. Finin, and B. Tseng, "Why We Twitter: Understanding Microblogging Usage and Communities," *Proc. Ninth WebKDD and First SNA-KDD Workshop Web Mining and Social Network Analysis (WebKDD/SNA-KDD '07)*, pp. 56-65, 2007.

[3] B. Huberman, D. Romero, and F. Wu, "Social Networks that Matter: Twitter Under the Microscope," *ArXiv E-Prints*, http://arxiv.org/abs/0812.1045, Dec. 2008.

[4] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, A Social Network or A News Media?" *Proc. 19th Int'l Conf. World Wide Web (WWW '10)*, pp. 591-600, 2010.

[5] G.L. Danah Boyd and S. Golder, "Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter," *Proc. 43rd Hawaii Int'l Conf. System Sciences (HICSS-43)*, 2010.

[6] A. Tumasjan, T.O. Sprenger, P.G. Sandner, and I.M. Welpe, "Predicting Elections with Twitter: What 140 Characters Reveal About Political Sentiment," *Proc. Fourth Int'l AAAI Conf. Weblogs and Social Media (ICWSM)*, 2010.

[7] P. Galagan, "Twitter as a Learning Tool. Really," *ASTD Learning Circuits*, p. 13, 2009.

[8] K. Borau, C. Ullrich, J. Feng, and R. Shen, "Microblogging for Language Learning: Using Twitter to Train Communicative and Cultural Competence," *Proc. Eighth Int'l Conf. Advances in Web Based Learning (ICWL '09)*, pp. 78-87, 2009.

[9] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer*, vol. 34, no. 8, pp. 57-66, 2001.

[10] M. Weiser, "The Computer for the Twenty-First Century," *Scientific Am.*, vol. 265, no. 3, pp. 94-104, 1991.

[11] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian Filtering for Location Estimation," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 24-33, July-Sept. 2003.

[12] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake Shakes Twitter Users: Real-Time Event Detection by Social Sensors," *Proc. 19th Int'l Conf. World Wide Web (WWW '10)*, pp. 851-860, 2010.

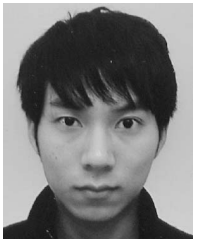
[13] Y. Raimond and S. Abdallah, "The Event Ontology," http://motools.sf.net/event/event.html, 2007.

[14] T. Joachims, "Text Categorization with Suport Vector Machines: Learning with Many Relevant Features," *Proc. 10th European Conf. Machine Learning (ECML '98)*, pp. 137-142, 1998.

[15] X. Liu, S. Zhang, F. Wei, and M. Zhou, "Recognizing Named Entities in Tweets," *Proc. 49th Ann. Meeting of the Assoc. for Computational Linguistics: Human Language Technologies (HLT '11)*, pp. 359-367, June 2011.

[16] A. Ritter, S. Clark Mausam, and O. Etzioni, "Named Entity Recognition in Tweets: An Experimental Study," *Proc. Conf. Empirical Methods in Natural Language Processing,* 2011.
[17] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Trans. Signal Processing,* vol. 50, no. 2, pp. 174-188, Feb. 2002.
[18] J. Leskovec, L.A. Adamic, and B.A. Huberman, "The Dynamics of Viral Marketing," *Proc. Seventh ACM Conf. Electronic Commerce (EC '06),* pp. 228-237, 2006.
[19] Y. Matsuo and H. Yamamoto, "Community Gravity: Measuring Bidirectional Effects by Trust and Rating on Online Social Networks," *Proc. 18th Int'l Conf. World Wide Web (WWW '09),* pp. 751-760, 2009.
[20] W. Zhu, C. Chen, and R.B. Allen, "Analyzing the Propagation of Influence and Concept Evolution in Enterprise Social Networks Through Centrality and Latent Semantic Analysis," *Proc. 12th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD '08),* pp. 1090-1098, 2008.
[21] E. Scordilis, C. Papazachos, G. Karakaisis, and V. Karakostas, "Accelerating Seismic Crustal Deformation before Strong Mainshocks in Adriatic and Its Importance for Earthquake Prediction," *J. Seismology,* vol. 8, pp. 57-70, http://dx.doi.org/10.1023/B:JOSE.0000009504.69449.48, 2004.
[22] T. Bleier and F. Freund, "Earthquake [earthquake warning systems]," *IEEE Spectrum,* vol. 42, no. 12, pp. 22-27, Dec. 2005.
[23] M. Cataldi, L. Di Caro, and C. Schifanella, "Emerging Topic Detection on Twitter Based on Temporal and Social Terms Evaluation," *Proc. 10th Int'l Workshop Multimedia Data Mining (MDMKDD '10),* pp. 1-10, 2010.
[24] B. O'Connor, R. Balasubramanyan, B.R. Routledge, and N.A. Smith, "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series," *Proc. Int'l AAAI Conf. Weblogs and Social Media,* 2010.
[25] M. Ebner and M. Schiefner, "Microblogging - More than Fun?" *Proc. IADIS Mobile Learning Conf.,* pp. 155-159, 2008.
[26] A. Passant, T. Hastrup, U. Bojars, and J. Breslin, "Microblogging: A Semantic Web and Distributed Approach," *Proc. Fourth Workshop Scripting for the Semantic Web (SFSW '08),* http://data.semanticweb.org/workshop/scripting/2008/paper/11, 2008.
[27] L. Backstrom, J. Kleinberg, R. Kumar, and J. Novak, "Spatial Variation in Search Engine Queries," *Proc. 17th Int'l Conf. World Wide Web (WWW '08),* pp. 357-366, 2008.
[28] Q. Mei, C. Liu, H. Su, and C. Zhai, "A Probabilistic Approach to Spatiotemporal Theme Pattern Mining on Weblogs," *Proc. 15th Int'l Conf. World Wide Web (WWW '06),* pp. 533-542, 2006.
[29] P. Serdyukov, V. Murdock, and R. van Zwol, "Placing Flickr Photos on a Map," *Proc. 32nd Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '09),* pp. 484-491, 2009.
[30] T. Rattenbury, N. Good, and M. Naaman, "Towards Automatic Extraction of Event and Place Semantics from Flickr Tags," *Proc. 30th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '07),* pp. 103-110, 2007.
[31] J. Hightower and G. Borriello, "Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study," *Proc. Int'l Conf. Ubiquitous Computing (UbiComp '04),* pp. 88-106, 2004.

Takeshi Sakaki received the BS and MS degrees from the University of Tokyo, Japan, in 2004 and 2006, respectively. Currently, he is working toward the PhD degree from the University of Tokyo, Japan. His research interests include natural language processing, Web mining, and artificial intelligence.



Makoto Okazaki received the BS degree from the University of Tokyo, Japan, in 2010. His research interests include Web mining, data mining, and artificial intelligence.

Yutaka Matsuo received the BS, MS, and PhD degrees from the University of Tokyo, in 1997, 1999, and 2002, respectively. He is an associate professor at the Institute of Engineering Innovation, The University of Tokyo, Japan. He joined National Institute of Advanced Industrial Science and Technology (AIST) from 2002 to 2007. He is interested in social network mining, text processing, and semantic web in the context of artificial intelligence research.