Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors

Takeshi Sakaki
The University of Tokyo
Yayoi 2-11-16, Bunkyo-ku
Tokyo, Japan
sakaki@biz-model.t.u-tokyo.ac.jp

Makoto Okazaki
The University of Tokyo
Yayoi 2-11-16, Bunkyo-ku
Tokyo, Japan
m_okazaki@biz-model.t.u-tokyo.ac.jp

Yutaka Matsuo
The University of Tokyo
Yayoi 2-11-16, Bunkyo-ku
Tokyo, Japan
matsuo@biz-model.t.u-tokyo.ac.jp

ABSTRACT

Twitter, a popular microblogging service, has received much attention recently. An important characteristic of Twitter is its real-time nature. For example, when an earthquake occurs, people make many Twitter posts (*tweets*) related to the earthquake, which enables detection of earthquake occurrence promptly, simply by observing the tweets. As described in this paper, 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 and the trajectory of the event location. We consider each Twitter user as a *sensor* and apply Kalman filtering and particle filtering, which are widely used for location estimation in ubiquitous/pervasive computing. The particle filter works better than other compared methods in estimating the centers of earthquakes and the trajectories of typhoons. As an application, we construct an earthquake reporting system in Japan. Because of the numerous earthquakes and the large number of Twitter users throughout the country, we can detect an earthquake by monitoring tweets with high probability (96% of earthquakes of Japan Meteorological Agency (JMA) seismic intensity scale 3 or more are detected). Our system detects earthquakes promptly and sends e-mails to registered users. Notification is delivered much faster than the announcements that are broadcast by the JMA.

1. INTRODUCTION

Twitter, a popular microblogging service, has received much attention recently. It is an online social network used by millions of people around the world to stay connected to their friends, family members and co-workers through their computers and mobile phones [18]. Twitter asks one question, "What are you doing?" 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; and her followers can read her tweets. A user who is being followed by another user need not necessarily have to reciprocate by following them back, which renders the links of the network as directed. After its launch on July 2006, Twitter users have increased rapidly. They are currently estimated as 44.5 million worldwide[1]. Monthly growth of users has been 1382% year-on-year, which makes Twitter one of the fastest-growing sites in the world[2].

Some studies have investigated Twitter: Java et al. analyzed Twitter as early as 2007. They described the social network of Twitter users and investigated the motivation of Twitter users [13]. B. 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 [11]. Recently, boyd et al. investigated *retweet* activity, which is the Twitter-equivalent of e-mail forwarding, where users post messages originally posted by others [5].

Twitter is categorized as a micro-blogging service. Microblogging is a form of blogging that allows users to send brief text updates or micromedia such as photographs or audio clips. Microblogging services other than Twitter include Tumblr, Plurk, Emote.in, Squeelr, Jaiku, identi.ca, and so on[3]. They have their own characteristics. Some examples are the following: Squeelr adds geolocation and pictures to microblogging, and Plurk has a timeline view integrating video and picture sharing. Although our study is applicable to other microblogging services, in this study, we specifically examine Twitter because of its popularity and data volume.

An important common characteristic among microblogging services is its 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. The large number of updates 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 storm, fire, traffic jam, riots, heavy rainfall, and earthquakes. Actually, Twitter is used for various real-time notification such as that necessary for help during a large-scale fire emergency and live traffic updates. Adam Ostrow, an Editor in Chief at Mashable, a social media news blog, wrote in his blog about the interesting phenomenon of the real-time media as follows[4]:

[1]http://www.techcrunch.com/2009/08/03/twitter-reaches-44.5-million-people-worldwide-in-june-comscore/
[2]According to a report from Nielsen.com.
[3]www.tumblr.com, www.plurk.com, www.emote.in, www.squeelr.com, www.jaiku.com, identi.ca
[4]http://mashable.com/2009/08/12/japan-earthquake/

WWW2010, April 26-30, 2010, Raleigh, North Carolina.
.

Japan Earthquake Shakes Twitter Users ... And Beyonce: Earthquakes are one thing you can bet on being covered on Twitter (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 and proposes an event notification system that monitors tweets and delivers notification promptly. 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" thus *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 based on features such as keywords in a tweet, the number of words, and the context of target-event words.

Subsequently, we make a probabilistic spatiotemporal model of an event. We make a crucial assumption: each Twitter user is regarded as a *sensor* and each tweet as *sensory information*. These virtual sensors, which we call *social sensors*, are of a huge variety and have various characteristics: some sensors are very active; others are not. A sensor could be inoperable or malfunctioning sometimes (e.g., a user is sleeping, or busy doing something). Consequently, social sensors are very noisy compared to ordinal physical sensors. Regarding a Twitter user as a sensor, the event detection problem can be reduced into the object detection and location estimation problem 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, 25]. We apply Kalman filters and particle filters, which are widely used for location estimation in ubiquitous/pervasive computing.

As an application, we develop an earthquake reporting system using Japanese tweets. Because of the numerous earthquakes in Japan and the numerous and geographically dispersed Twitter users throughout the country, it is sometimes possible to detect an earthquake by monitoring tweets. In other words, many earthquake events occur in Japan. Many sensors are allocated throughout the country. Figure 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, which means regions with many earthquakes and large Twitter users, is Japan. (Other regions such as Indonesia, Turkey, Iran, Italy, and Pacific US cities such as Los Angeles and San Francisco also roughly intersect, although the density is much lower than in Japan.) 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 has about 20 s before the arrival of an earthquake wave.

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 11th largest number of users (more than half a million users) in the world. Although event detection (particularly the earthquake detection) is currently possible because of the high density of Twitter users and earthquakes in Japan, our study is useful to detect events of various types throughout the world.

The contributions of the 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, which has not been reported before in the literature.

This paper is organized as follows: In the next section, we explain semantic analysis and sensory information, followed by the spatiotemporal model in Section 3. In Section 4, we describe the experiments and evaluation of event detection. The earthquake reporting system is introduced into Section 5. Section 6 is devoted to related works and discussion. Finally, we conclude the paper.

2. EVENT DETECTION

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 [21]. We target events such as earthquakes, typhoons, and traffic jams, which are visible through tweets. These events have several properties: i) they are of large scale (many users experience the event), ii) they particularly influence people's daily life (for that reason, they are induced to tweet about it), and iii) they have both spatial and temporal regions (so that real-time location estimation would be 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 rainfall, tornadoes, typhoons/hurricanes/cyclones, and earthquakes. We designate an event we would like to detect using Twitter as a *target event*.

2.1 Semantic Analysis on Tweet

To detect a target event from Twitter, we search from Twitter and find useful tweets. Tweets might include mentions of the target event. For example, users might make tweets such as "Earthquake!" or "Now it is shaking". Consequently, *earthquake* or *shaking* could be keywords (which we call *query words*). but 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

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

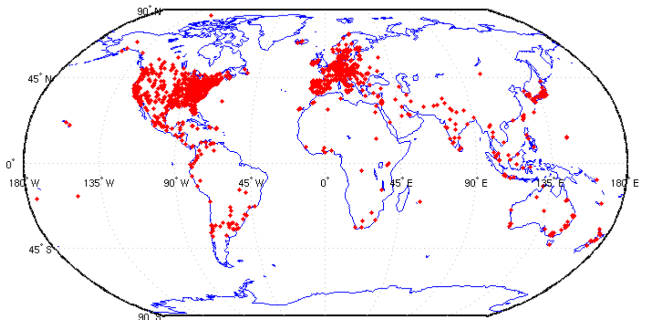


Figure 1: Twitter user map.

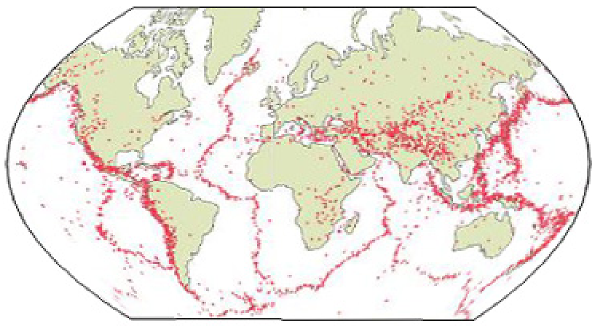


Figure 2: Earthquake map.

tweet is referring to the target event, it might not be appropriate as an event report; for example a user makes tweets such as "The earthquake yesterday was scaring", or "Three earthquakes in four days. Japan scares me." These tweets are truly the mentions of the target event, but they are not real-time reports of the events. Therefore, it is necessary to clarify that a tweet is actually referring to an actual earthquake occurrence, which is denoted as a positive class.

To classify a tweet into a positive class or a negative class, we use a support vector machine (SVM) [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.

We prepare three groups of features for each tweet as follows:

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.

To handle Japanese texts, morphological analysis is conducted using Mecab[7], which separates sentences into a set of words. In the case of English, we apply a standard stop-word elimination and stemming. We compare the usefulness of the features in Section 4. Using the obtained model, we can classify whether a new tweet corresponds to a positive class or a negative class.

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

2.2 Tweet as a Sensory Value

We can search the tweet and classify it into a positive class if a user makes a tweet on 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 considered as a *sensor reading*. This is a crucial assumption, but it enables application of various methods related to sensory information.

Assumption 2.1 *Each Twitter user is regarded as a sensor. A sensor detects a target event and makes a report probabilistically.*

The virtual sensors (or social sensors) have various characteristics: some sensors are activated (i.e. make tweets) only about specific events, although others are activated to a wider range of events. The number of sensors is large; there are more than 40 million sensors worldwide. A sensor might be inoperable or operating incorrectly sometimes (which means a user is not online, sleeping, or is busy doing something). Therefore, this social sensor is noisier than ordinal physical sensors such as location sensors, thermal sensors, and motion sensors.

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, e.g. 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 think it is probable that a person is near the registered location. In this study, we use GPS data and the registered location of a user. We do not use the tweet for spatial analysis if the location is not available (We use the tweet information for temporal analyses.).

Assumption 2.2 *Each tweet is associated with a time and location, which is a set of latitude and longitude.*

By regarding a tweet as a sensory value associated with a location information, the event detection problem is reduced to detecting an object and its location from sensor readings. Estimating an object's location is arguably the most fundamental sensing task in many ubiquitous and pervasive computing scenarios [7].

Figure 3 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 by a classifier. A probabilistic model is used to detect an event, as described in the next section.

3. MODEL

In order 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.

3.1 Temporal Model

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

First, we examine the actual data. Figures 4 and 5 respectively present the numbers tweets for two target events:

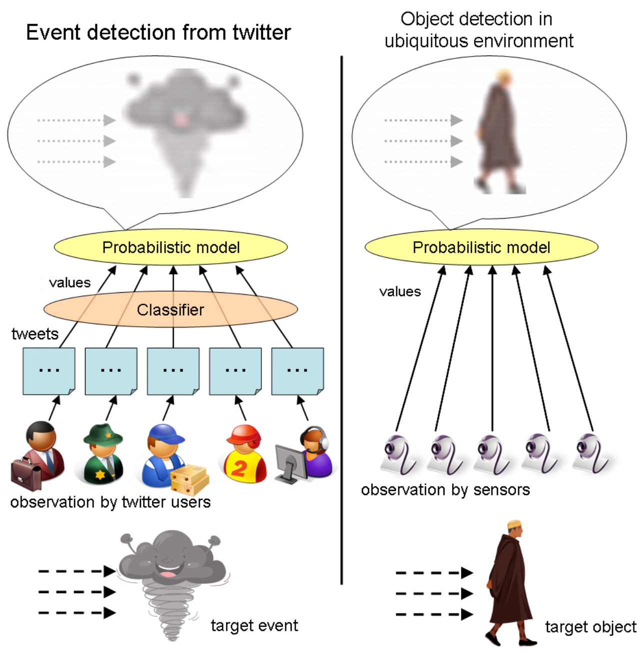


Figure 3: Correspondence between event detection from Twitter and object detection in a ubiquitous environment.

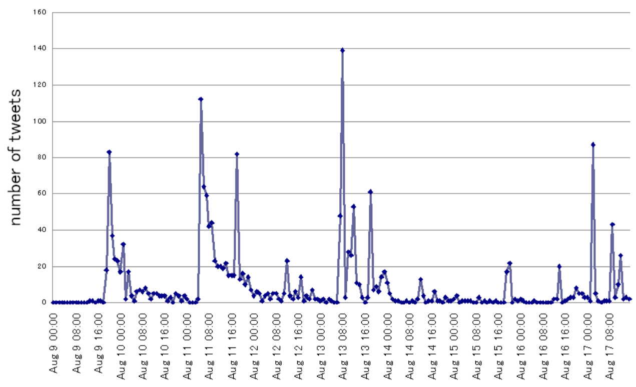


Figure 4: Number of tweets related to earthquakes.

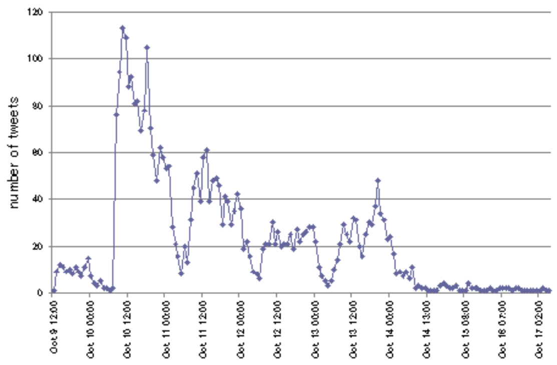


Figure 5: Number of tweets related to typhoons.

an earthquake and a typhoon. It is apparent that spikes occur on the number of tweets. Each corresponds to an event occurrence. In the case of an earthquake, more than 10 earthquakes occur during the period. In the case of typhoon, Japan's main population centers were hit by a large typhoon (designated as Melor) in October 2009.

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 inter-arrival times in a homogeneous Poisson process.

In the Twitter case, we can infer that if a user detects an event at time 0, assume that the probability of his posting a tweet from t to Δt is fixed as λ. Then, the time to make a tweet can be considered as an exponential distribution. Even if a user detects an event, therefore, she might not make a tweet right away if she is not online or doing something. 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 fits very well to an exponential distribution; we get $\lambda = 0.34$ with $R^2 = 0.87 on average$.

To assess an alarm, we must calculate the reliability of multiple sensor values. For example, a user might make a false alarm by writing a tweet. It is also possible that the classifier misclassifies a tweet into a positive class. 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 show in Section 4.1.
- Sensors are assumed to be independent and identically distributed (i.i.d.), as we explain in Section 3.3.

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 $n_0(1 - e^{-\lambda(t+1)})/(1 - e^{-\lambda})$. Consequently, the probability of an event occurrence at time t is

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

We can calculate the probability of event occurrence if we set $\lambda = 0.34$ and $p_f = 0.35$. For example, if we receive n_0 positive tweets and would like to make an alarm with a false-positive ratio less than 1%, we can calculate the expected wait time t_{wait} to deliver the notification as

$$t_{wait} = (1 - (0.1264/n_0))/0.7117 - 1.$$

Although many works describing event detection have been reported in the data mining field, we use this simple approach utilizing the characteristics of the classifier and the distribution.

3.2 Spatial Model

Each tweet is associated with a location. We describe how to 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 $x_t = f_t(x_{t-1}, v_{t-1})$, where $f_t : \mathcal{R}_t^n \times \mathcal{R}_t^n \rightarrow \mathcal{R}_t^n$ is a possibly nonlinear function of the state x_{t-1}. Furthermore, v_{t-1} is an i.i.d process noise sequence. The objective of tracking is to estimate x_t recursively from measurements $z_t = h_t(x_t, n_t)$, where $h_t : \mathcal{R}_t^n \times \mathcal{R}_t^n \rightarrow \mathcal{R}_t^n$ 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 $p(x_t|x_{t-1}, z_{t-1}) = p(x_t|x_{t-1})$. In update stage, the 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, multi-hypothesis tracking, grid-based and topological approaches, and particle filters [7]. For this study, we use Kalman filters and particle filters, both of which are widely used in location estimation.

3.2.1 Kalman Filters

The Kalman filter assumes that the posterior density at every time step is Gaussian and that it is therefore parameterized by a mean and covariance. We can write it as $x_t = F_t x_{t-1} + v_{t-1}$ and $z_t = H_t x_t + n_t$. Therein, F_k and H_k are known matrices defining the linear functions. The covariants of v_{k-1} and n_k are, respectively, Q_{t-1} and R_k.

The Kalman filter algorithm can consequently be viewed as the following recursive relation:

$$\begin{aligned} p(x_{t-1}|z_{t-1}) &= \mathcal{N}(x_{t-1}; m_{t-1|t-1}, P_{t-1|t-1}) \\ p(x_t|z_{t-1}) &= \mathcal{N}(x_t; m_{t|t-1}, P_{t|t-1}) \\ p(x_t|z_t) &= \mathcal{N}(x_t; m_{t|t}, P_{t|t}) \end{aligned}$$

where $m_{t|t-1} = F_t m_{t-1|t-1}$, $P_{t|t-1} = Q_{t-1} + F_t P_{t-1|t-1} F_t^T$, $m_{t|t} = m_{t|t-1} + K_t(z_t - H_t m_{t|t-1})$, and $P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}$, and where $\mathcal{N}(x; m, P)$ is a Gaussian density with argument x, mean m, covariance P, and for which the following are true: $K_t = P_{t|t-1} H_t^T S_t^{-1}$, and $S_t = H_t P_{t|t-1} H_t^T + R_t$. This is the optimal solution to the tracking problem if the assumptions hold. A Kalman filter works better in a linear Gaussian environment.

When utilizing Kalman filters, it is important to construct a good model and parameters. In this paper, we implement models for two cases as follows.

Case 1: Location estimation of an earthquake center. In this case, we need not take into consideration the time-transition property, thus we use only location information $x(d_x, d_y)$. We set $x_t = (d_{x_t}, d_{y_t})^t$ where d_{x_t} is the longitude and d_{y_t} is the latitude; $z_t = (d_{x_t}, d_{y_t})$, $F = I_2$, $H = I_2$, and $u_t = 0$. We assume that errors of temporal transition do not occur, and errors in observation are Gaussian for simplicity: $Q_t = 0$, $R_t = [\sigma^2]$, and $n_t = \mathcal{N}(0; R_t)$.

Case 2: Trajectory estimation of a typhoon. We need to consider both the location and the velocity of an event. We apply the Newton's motion equation as follows: $x_t = (d_{x_t}, d_{y_t}, v_{x_t}, v_{y_t})^t$ where v_{x_t} is the velocity on longitude, and v_{y_t} is the velocity on latitude. We set $z_t = (d_{x_t}, d_{y_t})^t$,
$F = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$, $H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$, $u_t = (\frac{a_{x_t}}{2}\Delta t^2, \frac{a_{y_t}}{2}\Delta t^2, a_{x_t}\Delta t, a_{y_t}\Delta t)^t$ where a_{x_t} is the acceleration on longitude, and a_{y_t} is the acceleration on latitude.

Similarly as in Case 1, we assume that errors of temporal transition do not occurr, and errors in observation are Gaussian for simplicity: $Q_t = 0$, $R_t = [\sigma^2]$, and $n_t = \mathcal{N}(0; R_t)$.

3.2.2 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 about the location of the object. The w_t^i are non-negative 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. We use a more advanced algorithm with re-sampling [1]. We employ weight distribution $D_w(x, y)$ which is obtained from twitter user distribution to take into consideration the biases of user locations[8] The alogorithm is shown in Algo. 1.

Algorithm 1 Particle filter algorithm

1. **Initialization**: Calculate the weight distribution $D_w(x, y)$ from twitter users geographic distribution in Japan.
2. **Generation**: Generate and weight a particle set, which means N discrete hypothesis.
   (1) Generate a particle set $S_0 = (s_{0,0}, s_{0,1}, s_{0,2}, \ldots, s_{0,N-1})$ and allocate them on the map evenly: particle $s_{0,k} = (x_{0,k}, y_{0,k}, weight_{0,k})$, where x corresponds to the longitude and y corresponds to the latitude.
   (2) Weight them based on weight distribution $D_w(x, y)$.
3. **Re-sampling**
   (1) Re-sample N particles from a particle set S_t using weights of each particles and allocate them on the map. (We allow to re-sample same particles more than one.)
   (2) Generate a new particle set S_{t+1} and weight them based on weight distribution $D_w(x, y)$.
4. **Prediction**: Predict the next state of a particle set S_t from the Newton's motion equation.

$$\begin{aligned} (x_{t,k}, y_{t,k}) &= (x_{t-1,k} + v_{x,t-1}\Delta t + \frac{a_{x,t-1}}{2}\Delta t^2, \\ &\quad y_{t-1,k} + v_{y,t-1}\Delta t + \frac{a_{y,t-1}}{2}\Delta t^2) \\ (v_{x,t}, v_{y,t}) &= (v_{x,t-1} + a_{x,t-1}, v_{y,t-1}, a_{y,t-1}) \end{aligned}$$

$$a_{x,t} = \mathcal{N}(0; \sigma^2), \quad a_{y,t} = \mathcal{N}(0; \sigma^2).$$

5. **Weighing**: Re-calculate the weight of S_t by measurement $m(m_x, m_y)$ as follows.

$$\begin{aligned} dx_k &= m_x - x_{t,k}, \quad dy_k = m_y - y_{t,k} \\ w_{t,k} &= D_w(x_{t,k}, y_{t,k}) \cdot \frac{1}{(\sqrt{2\pi}\sigma)} \cdot exp\left(-\frac{(dx_k^2 + dy_k^2)}{2\sigma^2}\right) \end{aligned}$$

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

3.3 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

[8]We sample tweets associated with locations and get user distribution proportional to the number of tweets in each region.

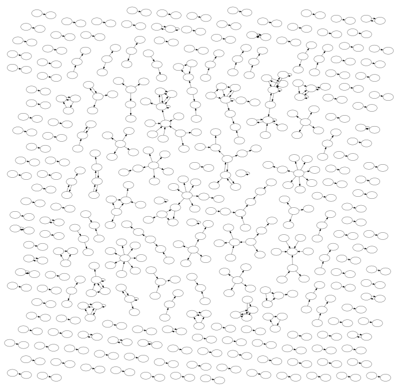


Figure 6: Earthquake information diffusion network.

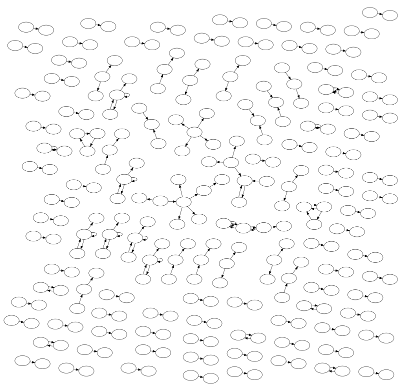


Figure 7: Typhoon information diffusion network.

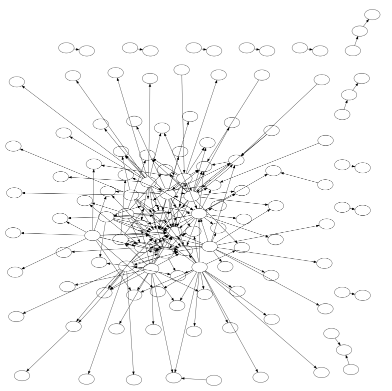


Figure 8: A new Nintendo game information diffusion network.

makes a tweet about the earthquake, a follower of that user might make tweets about that. This characteristic is important because, in our model, sensors might not be independent each other, which would cause an undesirable effect on event detection.

Figures 6, 7, and 8 respectively portray the information flow network on earthquake, typhoon, and a new Nintendo DS game[9]. We infer the network as follows: Assume that user A follows user B. If user B makes a tweet about an event, and soon after that if user A makes a tweet about an event, then we consider the information flows from B to A[10]. This is the similar definition to other studies of information diffusion (e.g., [15, 16]).

We can understand that, in the case of earthquakes and typhoons, very little information diffusion takes place on Twitter. On the other hand, the release of a new game illustrates the scale and rapidity of information diffusion. Therefore, we can assume that the sensors are i.i.d. when considering real-time event detection such as typhoons and earthquakes.

4. EXPERIMENTS AND EVALUATION

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

The whole algorithm is shown in Algo. 2. We prepare a set of queries Q for an 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 $Q = \{$"earthquake" and "shaking"$\}$ and 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. We consider that an event is detected if the probability is higher than a certain threshold ($p_{occur}(t) > 0.95$ in our case). The location information of each tweet is obtained and used for location estimation of the event. In the earthquake reporting system explained in the next section, the system quickly sends an e-mail (usually mobile e-mail) to registered users.

Algorithm 2 Event detection and location estimation algorithm.

```
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. Calculate event occurrence probability $p_{occur}$ using $v_t, t \in T$; if it is above the threshold $p_{occur}^{thre}$, then proceed to step 5.
5. For each tweet $t \in T$, we obtain the latitude and the longitude $l_t$ by i) utilizing the associated GPS location, ii) making a query to Google Map the registered location for user $u_t$. Set $l_t$ = null if both do not work.
6. Calculate the estimated location of the event from $l_t, t \in T$ using Kalman filtering or particle filtering.
7. (optionally) Send alert e-mails to registered users.
```

4.1 Evaluation by Semantic Analysis

For classification of tweets, we prepared 597 positive examples which report earthquake occurrence as a training set. The classification performance is presented in Table 1[12]. We use two query words—*earthquake* and *shaking*; performances using either query are shown. We used a linear kernel for SVM. We obtain the highest F-value when we use feature A and all features. Surprisingly, feature B and feature 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 so high as precision. It is attributable to the usage of query words in a different context than we intend. 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.

4.2 Evaluation of Spatial Estimation

Figure 9 presents the location estimation of an earthquake on August 11. We can find that many tweets originate from a wide region in Japan. The estimated location of the earthquake (shown as estimation by particle filter) is close to the actual center of the earthquake, which shows the efficiency of the location estimation algorithm. Table 2 presents re-

[9]Love Plus, a game that offers a virtual girlfriend experience, which was recently released in September 3, 2009.

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

[11]search.twitter.com

[12]We do not show the result for the typhoon case because of space limitations.

Table 1: Performance of classification.

(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%

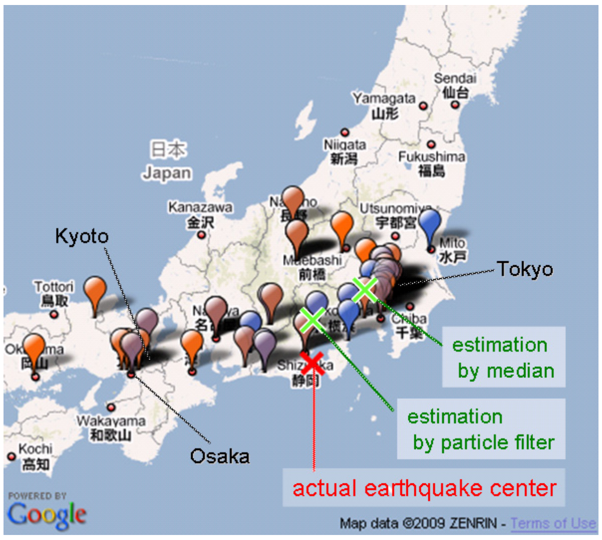


Figure 9: Earthquake location estimation based on tweets. Balloons show the tweets on the earthquake. The cross shows the earthquake center. Red represents early tweets; blue represents later tweets.

sults of location estimation for 25 earthquakes in August, September, and October 2009. We compare Kalman filtering and particle filtering, 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, and median simply takes the median of them. Particle filters perform well compared to other methods. The poor performance of Kalman filtering implies that the linear Gaussian assumption does not hold for this problem. We can find that if the center of the earthquake is in the sea area, it is more difficult to locate it precisely from tweets. Similarly, it becomes more difficult to make good estimations in less-populated areas. That is reasonable: all other things being equal, the greater the number of sensors, the more precise the estimation will be.

Figure 10 is the trajectory estimation of typhoon Melor based on tweets. In the case of an earthquake, the center is one location. However, in the case of a typhoon, the center moves and makes a trajectory. The comparison of the performance is shown in Table 3. The particle filter works well and outputs a similar trajectory to the actual trajectory.

5. EARTHQUAKE REPORTING SYSTEM

We developed an earthquake reporting system using the event detection algorithm. Earthquake information is much

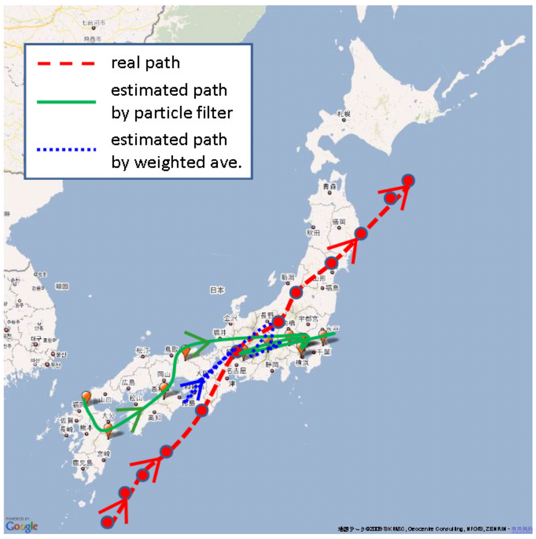

real path
estimated path by particle filter
estimated path by weighted ave.
Google

Figure 10: Typhoon trajectory estimation based on tweets.

more valuable if given in real time. We can turn off a stove or heater in our house and hide ourselves under a desk or table if we have several seconds before an earthquake actually hits. Several Twitter accounts report earthquake occurrence. Some examples are that the United States Geological Survey (USGS) feeds tweets on world earthquake information, but it is not useful for prediction or early warning.

Vast amounts of work have been done on intermediate-term earthquake prediction in the seismology field (e.g. [23]). Various attempts have also been made to produce short-term forecasts to realize an earthquake warning system by observing electromagnetic emissions from ground-based sensors and satellites [3]. Other precursor signals such as ionospheric changes, infrared luminescence, and air-conductivity change, along with traditional monitoring of movements of the earth's crust, are investigated.

In Japan, the government has allocated a considerable amount of its budget to mitigating earthquake damage. 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. It detects P-waves (primary waves) and makes an alert immediately so that earthquake damage can be mitigated through countermeasures such as slowing trains and controlling elevators. In fact, P-waves are a type of elastic wave that can travel faster than the S-waves (secondary waves), which cause shear effects and engender much more damage.

The proposed system, called *Toretter*[13], has been operated since August 8 of this year. A system screenshot is depicted in Fig. 11. Users can see the detection of past earthquakes. They can register their e-mails to receive notices of future earthquake detection reports. A sample e-mail is presented in Fig. 12. It alerts users and urges them to prepare for the earthquake. It is hoped that the e-mail is received by a user shortly before the earthquake actually arrives. An earthquake is transmitted through the earth's crust at about 3–7 km/s. Therefore, a person has about 20 s before its arrival at a point that is 100 km distant.

Table 4 presents some facts about earthquake detection and notification using our system. This table shows that we investigated 10 earthquakes during 18 August – 2 September, all of which our system detected. The first tweet of

[13]It means "we have taken it" in Japanese.

Table 2: Location estimation accuracy of earthquakes from tweets. For each method, we show the difference of the estimated latitude and the longitude to the actual ones, and the Euclid distance of them. Smaller distance means better performance.

Date	Actual center		Median (baseline)			Weighted ave. (baseline)			Kalman filters			**Particle filters**		
	lat.	long.	lat.	long.	dist.	lat.	long.	dist.	lat.	long.	dist.	lat.	long.	dist.
Aug. 10 01:00	33.10	138.50	3.40	-0.80	3.49	2.70	-0.10	2.70	2.67	-0.50	2.72	2.60	0.50	**2.65**
Aug. 11 05:00	34.80	138.50	0.90	-0.90	1.27	0.70	-0.30	0.76	0.60	-0.20	**0.63**	0.30	-0.90	0.95
Aug. 13 07:50	33.00	140.80	1.30	-9.60	9.69	2.30	-2.30	**3.25**	1.63	-3.75	4.09	2.70	-2.70	3.82
Aug. 17 20:40	33.70	130.20	4.60	6.00	7.56	0.90	3.20	3.32	1.63	4.35	4.65	0.10	-0.80	**0.81**
Aug. 18 22:17	23.30	123.50	7.80	9.90	12.60	8.70	10.90	13.95	8.32	10.13	13.11	5.60	8.10	**9.85**
Aug. 21 08.51	35.70	140.00	0.50	-4.40	4.43	0.10	-1.00	1.00	0.00	-0.60	**0.60**	-0.80	0.48	0.93
Aug. 24 13:30	37.50	138.60	-0.40	0.00	**0.40**	-0.50	0.40	0.64	-0.50	0.30	0.58	2.40	0.70	2.50
Aug. 24 14:40	41.10	140.30	-1.90	1.10	2.20	-1.30	0.50	**1.39**	-1.50	0.50	1.58	3.10	2.00	3.69
Aug. 25 02:22	42.10	142.80	-2.90	-3.90	4.86	-6.10	-3.80	7.19	-5.20	-3.70	6.38	-1.80	-1.90	**2.62**
Aug. 25 20:19	35.40	140.40	1.60	-1.80	2.41	2.20	-0.70	2.31	0.70	-1.60	1.75	1.40	0.10	**1.40**
Aug. 31 00:46	37.20	141.50	-0.40	-3.60	3.62	-1.10	-2.30	2.55	-1.30	-2.20	2.56	-0.30	-0.30	**0.42**
Aug. 31 21:11	33.40	130.90	-4.50	-3.60	5.76	0.50	2.10	2.16	0.70	1.90	2.02	-0.20	-1.70	**1.71**
Sep. 3 22:26	31.10	130.30	6.20	-0.10	6.20	4.00	5.00	6.40	4.90	7.20	8.71	2.40	2.10	**3.19**
Sep. 4 11:30	35.80	140.10	3.10	-1.70	3.54	0.20	-0.90	**0.92**	0.00	-1.00	1.00	0.80	1.40	1.61
Sep. 05 10:59	37.00	140.20	-2.70	-8.30	8.73	-1.40	-3.10	**3.40**	-1.30	-3.30	3.55	-2.10	-5.80	6.17
Sep. 08 01:24	42.20	143.00	-3.60	-8.90	9.60	-2.50	-3.90	4.63	-4.50	-6.00	7.50	1.30	-3.60	**3.83**
Sep. 10 18:29	43.20	146.20	-5.90	-10.20	11.78	-4.90	-7.10	8.63	-4.50	-7.20	8.49	-0.90	-7.00	**7.06**
Sep. 16 21:38	33.40	130.90	1.10	-0.20	**1.12**	0.90	2.10	2.28	0.50	1.40	1.49	-0.20	-2.50	2.51
Sep. 22 20:40	47.60	141.70	-11.10	-7.50	13.40	-10.80	-3.10	11.24	-11.30	-3.80	11.92	-7.80	-3.00	**8.36**
Oct. 1 19:43	36.40	140.70	0.70	-3.80	3.86	-0.60	-1.80	1.90	-0.30	-1.50	1.53	-0.70	0.30	**0.76**
Oct. 5 09:35	42.40	141.60	-3.70	-3.10	4.83	-2.70	-2.00	3.36	-2.60	-1.60	3.05	1.10	-1.70	**2.02**
Oct. 6 07:49	35.90	137.60	0.50	1.20	1.30	-0.20	0.80	0.82	-0.10	0.90	0.91	0.30	0.50	**0.58**
Oct. 10 17:43	41.80	142.20	-3.50	-5.40	6.44	-1.40	-2.10	**2.52**	-2.20	-2.60	3.41	2.40	-1.30	2.73
Oct. 12 16:10	35.90	137.60	2.80	0.50	2.84	0.80	1.20	**1.44**	0.80	1.60	1.79	3.60	1.40	3.86
Oct. 12 18:42	37.40	139.70	-2.00	-4.40	4.83	-1.50	-0.90	1.75	-1.70	-1.40	2.20	-1.00	-0.60	**1.17**
Average distance					5.47			3.62			3.85			**3.01**

Table 3: Trajectory estimation accuracy of typhoon Melor from tweets.

Date	Location		Median (baseline)			Weighted ave. (baseline)			Kalman filters			**Particle filters**		
	lat.	long.	lat.	long.	dist.	lat.	long.	dist.	lat.	long.	dist.	lat.	long.	dist.
Oct. 7 12:00	29.00	131.80	-1.90	-1.90	**2.69**	-5.20	-3.60	6.32	-3.90	-1.10	4.05	-4.70	1.10	4.83
Oct. 7 15:00	29.90	132.50	-3.70	-2.60	4.52	-3.80	-2.40	4.49	3.20	3.10	4.46	-2.70	0.90	**2.85**
Oct. 7 18:00	30.80	133.20	-4.10	-1.90	4.52	-4.40	-3.50	5.62	-6.40	5.40	8.37	-3.20	-0.70	**3.28**
Oct. 7 21:00	31.60	134.30	-3.90	-3.50	5.24	-3.60	-3.30	4.88	-10.90	-1.60	11.02	-3.70	-0.50	**3.73**
Oct. 8 0:00	32.90	135.60	-2.30	-0.10	**2.30**	-2.30	-0.90	2.47	-12.60	-20.40	23.98	-2.90	-3.50	4.55
Oct. 8 6:00	35.10	137.20	1.60	3.00	3.40	0.80	1.70	**1.88**	4.20	16.00	16.54	-0.60	-2.50	2.57
Oct. 8 9:00	36.10	138.80	-0.60	3.60	3.65	0.00	0.50	**0.50**	0.50	2.60	2.65	0.70	-0.80	1.06
Oct. 8 12:00	37.10	139.70	1.70	3.90	4.25	1.50	1.20	1.92	2.10	1.60	2.64	1.40	0.10	**1.40**
Oct. 8 15:00	38.00	140.90	2.30	3.20	3.94	2.40	2.20	**3.26**	1.70	7.60	7.79	2.40	2.70	3.61
Oct. 8 18:00	39.00	142.30	3.20	7.30	7.97	3.50	5.10	**6.19**	2.10	-18.80	18.92	3.70	5.10	6.30
Oct. 8 21:00	40.00	143.60	4.30	3.90	5.81	4.00	5.30	6.64	1.60	4.50	4.78	4.20	3.10	**5.22**
Average distance					4.39			4.02			9.56			**3.58**

Table 5: Earthquake detection performance for two months from August 2009.

JMA intensity scale	2 or more	3 or more	4 or more
Num. of earthquakes	78	25	3
Detected	70(89.7%)	24 (96.0%)	3 (100.0%)
Promptly detected[14]	53 (67.9%)	20 (80.0%)	3 (100.0%)

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 by our system. We apply classification for 49,314 tweets retrieved by query words in one month; results show 6,291 positive tweets posted by 4,218 users. Every earthquake elicited more than 10 tweets within 10 min, except one in Bungo-suido, which is the sea between two large islands: Kyushu and Shikoku. Our system sent e-mails mostly within a minute, sometimes within 20 s. The delivery time is far faster than the rapid broadcast of announcement of JMA, which are widely broadcast on TV; on average, a JMA announcement is broadcast 6 min after an earthquake occurs. Statistically, we detected 96% of earthquakes larger than JMA seismic intensity scale[15] 3 or more as shown in Table 5.

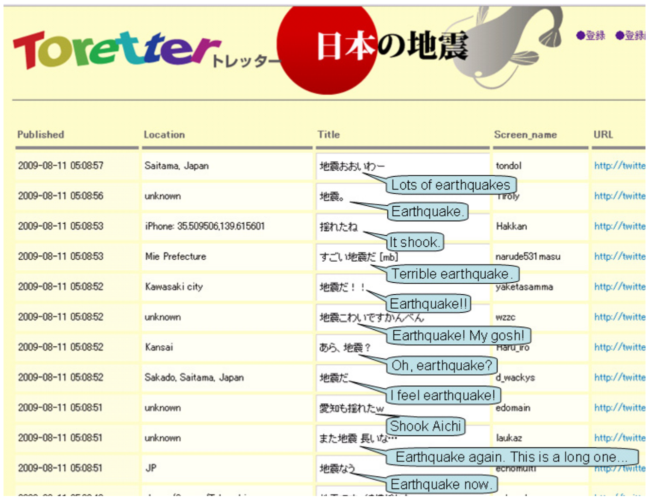
6. RELATED WORK

Twitter is an interesting example of the most recent social media: numerous studies have investigated Twitter. Aside from the studies introduced in Section 1, several others have been done. Grosseck et al. investigated indicators such as the influence and trust related to Twitter [8]. Krishnamurthy et al. crawled nearly 100,000 Twitter users and examined the number of users each user follows, in addition to the number of users following them. Naaman et al. analyzed contents of messages from more than 350 Twitter

[15]The JMA seismic intensity scale is a measure used in Japan and Taiwan to indicate earthquake strength. Unlike the Richter magnitude scale, the JMA scale describes the degree of shaking at a point on the earth's surface. For example, the JMA scale 3 is, by definition, one which is "felt by most people in the building. Some people are frightened". It is similar to the Modified Mercalli scale IV, which is used along with the Richter scale in the US.

Table 4: **Facts about earthquake detection.**

Date	Magnitude	Location	Time	E-mail sent time	#tweets within 10 min	Announce of JMA
Aug. 18	4.5	Tochigi	6:58:55	7:00:30	35	07:08
Aug. 18	3.1	Suruga-wan	19:22:48	19:23:14	17	19:28
Aug. 21	4.1	Chiba	8:51:16	8:51:35	52	8:56
Aug. 25	4.3	Uraga-oki	2:22:49	2:23:21	23	02:27
Aug. 25	3.5	Fukushima	22:21:16	22:22:29	13	22:26
Aug. 27	3.9	Wakayama	17:47:30	17:48:11	16	17:53
Aug. 27	2.8	Suruga-wan	20:26:23	20:26:45	14	20:31
Aug. 31	4.5	Fukushima	00:45:54	00:46:24	32	00:51
Sep. 2	3.3	Suruga-wan	13:04:45	13:05:04	18	13:10
Sep. 2	3.6	Bungo-suido	17:37:53	17:38:27	3	17:43



Published	Location	Title	Screen_name	URL
2009-08-11 05:08:57	Saitama, Japan	地震おおいわー	tondol	http://twitte
2009-08-11 05:08:56	unknown	地震。	Tiroly	http://twitte
2009-08-11 05:08:53	iPhone: 35.509506,139.615601	揺れたね	Hakkan	http://twitte
2009-08-11 05:08:53	Mie Prefecture	すごい地震だ [mb]	narude531masu	http://twitte
2009-08-11 05:08:52	Kawasaki city	地震だ！！	yaketasamma	http://twitte
2009-08-11 05:08:52	unknown	地震こわいですかんべん	wzzc	http://twitte
2009-08-11 05:08:52	Kansai	あら、地震？	Haru_iro	http://twitte
2009-08-11 05:08:52	Sakado, Saitama, Japan	地震だ	d_wackys	http://twitte
2009-08-11 05:08:51	unknown	愛知も揺れたw	edomain	http://twitte
2009-08-11 05:08:51	unknown	また地震 長いな…	laukaz	http://twitte
2009-08-11 05:08:51	JP	地震なう	echomulti	http://twitte

Figure 11: Screenshot of Toretter, an earthquake reporting system.

```
Dear Alice,

We have just detected an earthquake
around Chiba. Please take care.

Toretter Alert System
```

Figure 12: Sample alert e-mail.

users and manually classified messages into nine categories [19]. The numerous categories are "Me now" and "Statements and Random Thoughts"; statements about current events corresponding to this category.

Some studies attempt to show applications of Twitter: Borau et al. tried to use Twitter to teach English to English-language learners [4]. Ebner et al. investigated the applicability of Twitter for educational purposes, i.e. mobile learning [6]. The integration of the Semantic Web and microblogging was described in a previous study [20] in which a distributed architecture is proposed and the contents are aggregated. Jensen et al. analyzed more than 150 thousand tweets, particularly those mentioning brands in corporate accounts [12].

In contrast to the small number of academic studies of Twitter, many Twitter applications exist. Some are used for analyses of Twitter data. For example, Tweettronics[16] provides an analysis of tweets related to brands and products for marketing purposes. It can classify positive and negative tweets, and can identify influential users. The classification of tweets might be done similarly to our algorithm. Web2express Digest[17] is a website that auto-discovers 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 been made of the analysis of web data (except for Twitter) particularly addressing the spatial aspect: The most relevant study to ours is one by Backstrom et al. [2]. They use queries with location (obtained by IP addresses), and develop a probabilistic framework for quantifying spatial variation. The model is based on a decomposition of the surface of the earth into small grid cells; they assume that for each grid cell x, there is a probability p_x that a random search from this cell will be equal to the query under consideration. The framework finds a query's geographic center and spatial dispersion. Examples include baseball teams, newspapers, universities, and typhoons. Although the motivation is very similar, 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 [17]. They presented examples for Hurricane Katrina, Hurricane Rita, and iPod Nano. The motivation of that study is similar to ours, but Twitter data are more time-sensitive; our study examines even more time-critical events e.g. earthquakes.

Some works have targeted collaborative bookmarking data, as Flickr does, from a spatiotemporal perspective: Serdyukov et al. investigated generic methods for placing photographs on Flickr on the world map [24]. They used a language model to place photos, and showed that they can effectively estimate the language model through analyses of annotations by users. Rattenbury et al. [22] specifically examined the problem of extracting place and event semantics for tags that are assigned to photographs on Flickr. They proposed 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 statistically enables a unified interface for location information, which enables us to make applications independent of the sensors used — even when using very different sensor types, such as GPS and infrared badges [7], or even Twitter. Well known algorithms for location estimation are Kalman filters, multihypothesis tracking, grid-based, and topological approaches, and particle filters. Hightower and Borriello made a study of applying particle filters to location sensors deployed throughout a lab building [10]. More than

[16]http://www.tweettronics.com

[17]http://web2express.org

30 lab residents were tracked; their locations were estimated accurately using the particle filter approach.

7. DISCUSSION

We plan to expand our system to detect events of various kinds using Twitter. We developed another prototype that detects rainbow information. A rainbow might be visible somewhere in the world; someone might be twittering about a rainbow. Our system can identify rainbow tweets using a similar approach to that used for detecting earthquakes. The differences are that in the rainbow case, the information is not so time-sensitive as that in the earthquake case.

Our model includes the assumption that a single instance of the target event exists. For example, we assume that we do not have two or more earthquakes or typhoons simultaneously. Although the 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 allow hypotheses of multiple event occurrences.

A search query is important to search possibly-relevant tweets. For example, we set a query term 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. We can use advanced algorithms for query expansion, which is a subject of our future work.

8. CONCLUSION

As described in this paper, we investigated the real-time nature of Twitter, in particular for event detection. Semantic analyses were applied to tweets to classify them into a positive and a negative class. We consider each Twitter user as a sensor, and set a problem to detect an event based on sensory observations. Location estimation methods such as Kalman filtering and 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. In this paper, we presented an example using the real-time nature of Twitter. It is hoped that this paper provides some insight into the future integration of semantic analysis with microblogging data.

9. REFERENCES

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 2001.

[2] L. Backstrom, J. Kleinberg, R. Kumar, and J. Novak. Spatial variation in search engine queries. In *Proc. WWW2008*, 2008.

[3] T. Bleier and F. Freund. Earthquake warning system. *Spectrum, IEEE*, 2005.

[4] K. Borau, C. Ullrich, J. Feng, and R. Shen. Microblogging for language learning: Using twitter to train communicative and cultural competence. In *Proc. ICWL 2009*, pages 78–87, 2009.

[5] d. boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *Proc. HICSS-43*, 2010.

[6] M. Ebner and M. Schiefner. In microblogging.more than fun? In *Proc. IADIS Mobile Learning Conference*, 2008.

[7] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filters for location estimation. *IEEE Pervasive Computing*, 2003.

[8] G. Grosseck and C. Holotescu. Analysis indicators for communities on microblogging platforms. In *Proc. eLSE Conference*, 2009.

[9] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.

[10] J. Hightower and G. Borriello. Particle filters for location estimation in ubiquitous computing: A case study. In *Proc. UbiComp04*, 2004.

[11] B. Huberman and D. R. F. Wu. Social networks that matter: Twitter under the microscope. *First Monday*, 14, 2009.

[12] B. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power:tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, 2009.

[13] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: Understanding microblogging usage and communities. In *Proc. Joint 9th WEBKDD and 1st SNA-KDD Workshop 2007*, 2007.

[14] T. Joachims. Text categorization with support vector machines. In *Proc. ECML'98*, pages 137–142, 1998.

[15] J. Leskovec, L. Adamic, and B. Huberman. The dynamics of viral marketing. In *Proc. ACM Conference on Electronic Commerce*, 2006.

[16] Y. Matsuo and H. Yamamoto. Community gravity: measuring bidirectional effects by trust and rating on online social networks. In *Proc. WWW2009*, 2009.

[17] Q. Mei, C. Liu, H. Su, and C. Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proc. WWW'06*, 2006.

[18] S. Milstein, A. Chowdhury, G. Hochmuth, B. Lorica, and R. Magoulas. *Twitter and the micro-messaging revolution: Communication,connections, and immediacy.140 characters at a time.* O'Reilly Media, 2008.

[19] M. Naaman, J. Boase, and C. Lai. Is it really about me? Message content in social awareness streams. In *Proc. CSCW'09*, 2009.

[20] A. Passant, T. Hastrup, U. Bojars, and J. Breslin. Microblogging: A semantic and distributed approach. In *Proc. SFSW2008*, 2008.

[21] Y. Raimond and S. Abdallah. The event ontology, 2007. http://motools.sf.net/event/event.html.

[22] T. Rattenbury, N. Good, and M. Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *Proc. SIGIR 2007*, 2007.

[23] E. Scordilis, C. Papazachos, G. Karakaisis, and V. Karakostas. Accelerating sesmic crustal deformation before strong mainshocks in adriatic and its importance for earthquake prediction. *Journal of Seismology*, 8, 2004.

[24] P. Serdyukov, V. Murdock, and R. van Zwol. Placing flickr photos on a map. In *Proc. SIGIR 2009*, 2009.

[25] M. Weiser. The computer for the twenty-first century. *Scientific American*, 268(3):94–104, 1991.