〔研究ノート〕

# Topic Extraction from Two Hundred Million Tweets related to the East Japan Great Earthquake

HASHIMOTO, Takako

**Abstract**

Social media offers a wealth of insight into how significant topics—such as the Great East Japan Earthquake, the Arab Spring, and the Boston Bombing—affect individuals. The scale of available data, however, can be intimidating: during the Great East Japan Earthquake, over 8 million tweets were sent each day from Japan alone. Conventional word vector-based social media analysis method using Latent Semantic Analysis, Latent Dirichlet Allocation, or graph community detection often cannot scale to such a large volume of data due to their space and time complexity. To overcome the scalability problem, in this paper, both the method using high performance Singular Vector Decomposition (SVD) and the method using the original fast feature selection algorithm named CWC are introduced. We target the huge data set of over two hundred million tweets sent in the 21 days following the Great East Japan Earthquake and begin with word count vectors of authors and words for each time slot (in our case, every hour). In the first method, authors' clusters from each slot are extracted by SVD and $k$-means. And then, the original fast feature selection algorithm named CWC has been used to extract discriminative words from each cluster. In the second method, we directly extract discriminative words from each slot using CWC. We then convert word vectors into a time series of vector distances to identify topics over time.

The first method still shows problems for topic extraction from big data. However, the second method can make it possible to detect events from vast datasets. From the experiment, though the emergent topics can be observed from the authors' clusters, the issues of conventional topic detection techniques from big data can also be identified as well.

## I. Introduction

Social media offers a wealth of insight into how significant topics –such as the Great East Japan Earthquake, the Arab Spring, and the Boston Bombing–affect individuals. The scale of available data, however, can be intimidating: during the Great East Japan Earthquake, over 8 million tweets per day were sent from Japan alone. Discovering such an event, and classifying tweets relevant to the event, remains an ongoing area of research. Many techniques such as graph based methods [1], Latent Semantic Analysis (LSA) [2] and Latent Dirichlet Allocation (LDA) [3] have been proposed so far, but none of them scales adequately to millions of tweets. To overcome the sociability problems, we already developed topic extraction methods [4] [5] from big data using the original technique CWC [6]. In this paper, our two methods are introduced. The first method [4] uses high performance Singular Vector Decomposition (SVD) to identify topic clusters over time from the huge data set of over two hundred million tweets sent in the 21 days following the Great East Japan Earthquake, and to confirm the feasibility of topic extraction from big data. Then, CWC [6], a fast feature selection technique is used to extract discriminative words from the clusters. The second method [5] directly extracts discriminative words from each slot using CWC. We then convert word vectors into a time series of vector distances to identify topics over time. The first method still shows problems for topic extraction from big data. However, the second method can make it possible to detect events from vast datasets.

The main contributions in the work [4] [5] are as follows:

- to improve the conventional social media analysis method for big data using high performance SVD library and the original fast feature selection technique CWC.
- to propose the original method to detect topics from vast datasets directly using CWC.
- to identify topics after the Great East Japan Earthquake from large twitter data.

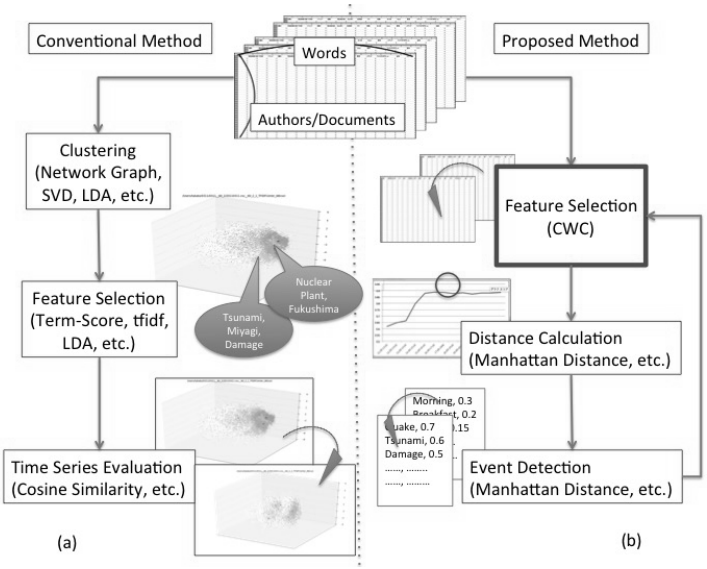


Fig. 1. Conventional Method (a) vs. Proposed Method (b)

- to discuss issues of conventional social media analysis method for big data.

We already developed the time series social media analysis technique for blog data related to the Great East Japan Earthquake [7]. But our previous technique targeted just around one thousand blog data. This work targets over 200 million Tweets, so that we have to develop new method for big data.

The paper is organized as follows. Section II introduces related work on social media analysis. Section III introduces our two methods using high performance SVD and the original feature selection technique CWC [4], [5]. Section IV demonstrates experimental results of our method. Section V discusses issues on the conventional social media analysis method. Finally, Section VI concludes this paper and offers directions for future research.

## II. Related Work

Most social media analysis methods comprise of the following basic template (Figure 1 (a)):

1) Form matrices (or bipartite graphs) of connections between authors (or documents) and words over time.
2) For each matrix, form clusters and adopt a topic modeling technique such as LDA, or $k$-means [9] algorithm with dimensionality reduction such as LSA or adopt a network community extraction method in case of bipartite graphs.
3) For each cluster, define important keywords to represent the contents (LDA also produces keyword importance scores)

Generally, this conventional method lacks scalability. Existing data mining technique target thousands of items, not millions. For example, Fujino et al. [10] analyzed tweets over time based on LDA, but the number of their targeted tweets was only around 200K. Paul et al. [11] proposed a topic model based on LDA and targeted over 100 million tweets. However, they had to filter them first to reduce data until it reached to appropriate data size (around 5000 tweets). Zhao et al. [12] analyzed twitter and news article using LDA. At first, the number of targeted tweets was 1 million, but they also filtered the data to reduce

its size. Kitada et al. [13] targeted 200 million tweets related to the Great East Japan Earthquake, and tried to analyze them by LDA based technique. However, they employed parallel processing to tackle big data. parallel processing is one of the solutions for handling big data, but to make big data analysis easier, high performance data mining technique is quite necessary.

This conventional method has several problems. First, existing data mining technique such as graph based methods, LSA and LDA target thousands of items, not millions. Second, in addition to lack of scalability, the accuracy of clustering (decomposition) techniques is not high, nor can these techniques deliver reasonable performance. Third, to extract important keywords from clusters, we generally use word scoring methods such as TF-IDF [14] or term-score [20]. However, such scoring methods are based on word occurrence, and high-frequency words tend to be extracted. Therefore, word scoring methods cannot always explain each cluster with high precision. Finally, sometimes these methods identify false similarities between clusters over time.

## III. Proposed Method

### A. Topic Extraction Using High Performance Singular Vector Decomposition

The first method that was already published in the paper [4], follows conventional method as well, but to scale to big data, high performance SVD library $redsvd$ [8] is employed for clustering and CWC is used for feature selection.

*1) Step 1: Creation of Author-Word Count Matrices:* In the first step, following conventional methods, the tweets are grouped by a certain period (e.g. hour) during which they were sent. Then the sequence of author-word count matrices , $\langle A_0, A_1 \ldots, A_t, \ldots, A_T \rangle$ that summarizes the words used in tweets by each author during each time slot are created.

$$A_t = \begin{pmatrix} a_{11} & a_{12} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{22} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{32} & \cdots & a_{3n} \\ & \vdots & & \ddots & \\ a_{m1} & a_{m2} & a_{m2} & \cdots & a_{mn} \end{pmatrix} = (a_{ij})_t$$

where $1 \leq i \leq m$ and $1 \leq j \leq n$. The index $m$ is the number of authors and $n$ is the number of words during a time period. The element $a_{ij}$ shows the number of times the $i$-th author used a particular word $w_j$ during a time period. These time series matrices, $A_0, \ldots, A_T$, are obviously sparse. We assume that any significant event does not happen in the first time period $t = 0$, and let $A_0$ be the initial matrix representing an ordinary state.

*2) Step 2: Clustering:* We calculate TF-IDF [14] for $(a_{ij})_t$ and apply $redsvd$ for reducing dimensions of each author-word matrix. $redsvd$ is $C++$ library for solving several matrix decompositions. It can handle very large matrix efficiently, and is optimized for a truncated SVD of sparse matrices. For example, $redsvd$ can compute a truncated SVD with top 20 singular values for a 100K x 100K matrix with 1M nonzero entries in less than one second.

Truncated SVD's formula is as follows:

$$A \approx U_r \Sigma_r V_r^T$$

where $U_r$ is an $m \times r$ matrix of authors, $\Sigma_r$ is an $m \times r$ rectangular diagonal matrix, and $V_r^T$ is an $r \times n$ matrix of words. By setting a specific rank $r$, $A$ is approximated as $U_r \Sigma_r V_r^T$. Only the $r$ column vectors of $U$ and $r$ row vectors of $V^T$ corresponding to the $r$ largest singular values $\Sigma_r$ are calculated.

Then a matrix of the first main component to the n-th main component from $U_r$ is obtained and clusters are formed by $k$-means, each cluster shows a group of authors.

TABLE I
AN EXAMPLE OF DATASET

| $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $C$ |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |

*3) Step 3: Feature Selection :* For clusters of each time slot, the fast feature selection algorithm CWC is applied.

CWC is an accurate and fast feature selection algorithm for categorical data. Feature selection addresses the problem of finding a small set of features relevant to class labels. Table I shows an example of a dataset (note that CWC can deal with multi-category in general, but we use two category problem here for simplicity). The features are denoted by $F_1, \ldots, F_5$, respectively, and the variable of the class labels for instances is denoted by $C$.

The single feature $F_2$ is useless to determine the class label since mutual information $I(F_2, C) = 0$. In the same way, the single feature $F_5$ is also useless due to $I(F_5, C) = 0$. In contrast, the single feature $F_4$ is more informative than $F_2$ and $F_5$ to determine the class label since $I(F_4, C) = 0.13$. Let us consider the combination of features $F_2$ and $F_5$. Then, these features completely determine the class label since $I(\{F_2, F_5\}, C) = 1$, and the negation of exclusive-OR of $F_2$ and $F_5$ is equivalent to $C$.

This example suggests that it is essential to search for combination of features relevant to class labels. The most prospective method to address the problem is called *consistency-based feature selection* [15]. If a subset of features is *consistent*, it implies that the subset completely determines all the class labels.

CWC is one of the fastest consistency-based feature selection algorithms. CWC employs the simplest consistency measure for the criteria of feature selection called *binary consistency measure*. This measure just discriminates whether the subset of features can completely determine all the class labels or not. Recently, we have further improved CWC by incorporating a drastically faster search strategy and adapting it to sparse datasets for handling a massive amount of data.

### *B. Topic Extraction Using High Performance Feature Selection Technique*

In contrast to these conventional methods, we proposed a method for detecting events from huge amounts of social media using feature selection (Figure 1 (b)) [5] . This section will present our new technique. This method offers better performance and accuracy than the previously-discussed methods, and will scale well to big data.

Our method consists of the following 4 steps (Figure 2):

*1) Step 1: Creation of Author-Word Count Matrices:* First, following conventional methods, we group the tweets by a certain period (e.g. hour) during which they were sent. We then create the sequence of author-word count matrices, $\langle A_0, A_1 \ldots, A_t, \ldots, A_T \rangle$ that summarizes the words used in tweets by each

author during each time slot.

$$A_t = \begin{pmatrix} a_{11} & a_{12} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{22} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{32} & \cdots & a_{3n} \\ & \vdots & & \ddots & \\ a_{m1} & a_{m2} & a_{m2} & \cdots & a_{mn} \end{pmatrix} = (a_{ij})_t$$

where $1 \leq i \leq m$ and $1 \leq j \leq n$. The index $m$ is the number of authors and $n$ is the number of words during a time period. The element $a_{ij}$ shows the number of times that $i$-th author used a particular word $w_j$ during a time period. These time series matrices, $A_0, \ldots, A_T$, are obviously sparse. We assume that any significant event does not happen in the first time period $t = 0$, and let $A_0$ be the initial matrix representing an ordinary state.

*2) Step 2: Apply Feature Selection:* Next, we apply a feature selection technique CWC to extracting the most discriminative set of words between a time slot's matrix $A_t (1 \leq t \leq T)$ and the initial matrix $A_0$. Let the set extracted words be

$$W_t = \{w_1^{(t)}, \ldots, w_{n_t}^{(t)}\}.$$

We call $W_t$ *principal word vector* at $t$. To each word $w_i^{(t)}$, we assign a score according to its discriminative relevance to the time period $t$ compared to the initial time period. We employ the *Matthew's Correlation Coefficient* (MCC) [19] for the score, which ranges from $-1$ to 1. We define $score_t(w)$ as the MCC value of word $w$ at $t$ compared to the first time period if $w \in W_t$, otherwise 0. We define $W_1$ at $t = 1$ as the initial principal word vector.

*3) Step 3: Distance Calculation:* We calculate the the Manhattan Distance [21] between each principal word vector $W_k (2 <= k <= T)$ and the initial principal word vector $W_1$ as follows (See Figure 2).

$$d(k, 1) = \sum_{w \in W_k \cup W_1} \left| score_k(w) - score_1(w) \right|.$$

This means that the distance from the initial principal word vector is calculated as the distance from an ordinary state.

*4) Step 4: Event Detection:* At Step 3, we compute the Manhattan Distance of every principal word vector to every subsequent time slot's word vector, yielding a set of time series of decreasing length. This time series, which we call a Manhattan Distance time series, shows each principal word vector's relative strength over time: in other words, how long each event lasts. The Manhattan Distance between each time slot and the initial principal word vector remains quite high if an event is happening, yet declines sharply when the event ends.

While line graphs of the Manhattan Distance time series make visual identification of events by a human being comparatively easy, the scale of our dataset makes human reading impractical. We can then extract the principal word vectors for each burst to characterize the event for human readability. If there is a big change, and after that, there is a stable line of the graph, we apply the feature selection technique for certain time slots again by shifting the initial matrix. This is a sort of iteration process to analyze the event deeply. To detect these events, we also plan to apply Kleinberg's burst-detection algorithm [17] to each Manhattan Distance time series, which yields the start and end of each burst. The resulting bursts are events.

Using burst detection overcomes some disadvantages of clustering algorithms. For example, $k$-means [9] clustering and LDA require the number of clusters to be pre-selected, while burst detection can detect an arbitrary number of bursts. Furthermore, as mentioned above, the accuracy of clustering algorithms is not as high as the accuracy of burst detection algorithms, given the arbitrary nature of the input number

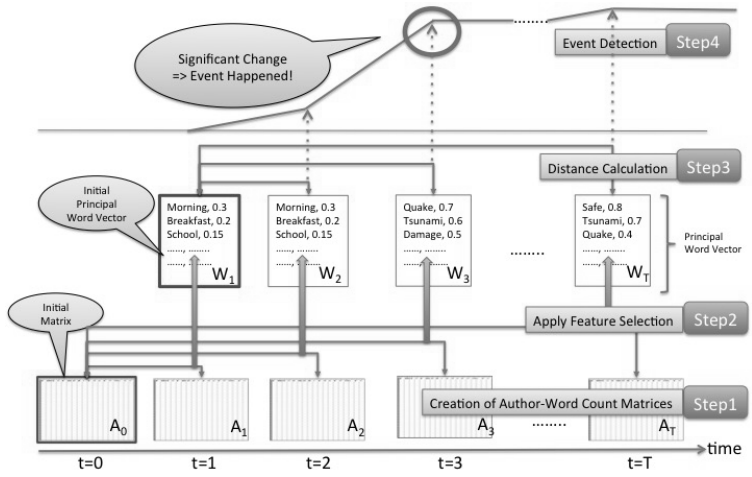


Fig. 2. Steps of Our Proposed Method

of clusters. Finally, the performance of burst detection algorithms is better since burst detection can be made linearly proportional to the input and output [22] and is well-suited for big data.

## IV. Experimental Result

In this section, our experimental results are reported. The experiment is conducted on the MabBook Air 1.7 GHz Core i7 with 8GB memory.

### *A. Target Data*

Our target data is over 200 million tweets in Japanese that were sent around the time of the Great East Japan Earthquake, starting from March 9, 2011. The social media monitoring company Hottolink [16] tracked users who used one of 43 hashtags (for example, #jishin, #nhk, and #prayforjapan) or one of 21 keywords related to the disaster. Later, they captured all tweets sent by all of these users between March 9th and March 29th. This resulted in an archive of around 200 million tweets, sent by around 1 million users. An average of about 8 million tweets were posted by around 200 thousand authors per day. The average data size per day was around 8GB, and the total data size was over 150GB. (Figure 3). This dataset offers a significant document of users' responses to a crisis, but its size presents a challenge for analysis.

In the following subsections, our experimental result for tweets from 9:00 on March 11 to 24:00 on March 12, a total of 39 hours are shown.

### *B. Creation of Author-Word Count Matrices*

In the first step of both methods, author-word count matrices are created from the dataset. The fast and customizable Japanese morphological analyzer, MeCab [18] is employed to segment tweets not having spaces to delineate word boundaries,. Author-word count matrices are created for a duration of one hour,e.g, each matrix for an hour on March 11 after 15:00 (the time of the earthquake), contains 600,000-980,000 tweets by 140,000-165,000 authors with over 200,000 words. The total size of each matrix is over 30MB and they were all quite sparse.

Table II shows the exact number of authors, words, and total size of each hour's matrix derived from tweets on March 11, 2011.

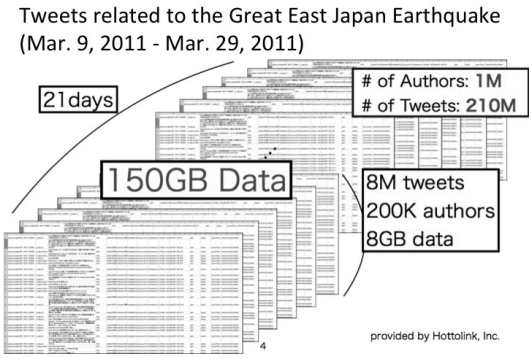


Fig. 3. Target Data: 200 million tweets related to the Great East Japan Earthquake

TABLE II
AUTHOR-WORD MATRICES ON MAR. 11

| hour (24h) | # of tweets | # of authors | # of words | size of file (MB) |
|---|---|---|---|---|
| 09 - 10 | 136167 | 48711 | 147271 | 4.6 |
| 10 - 11 | 138491 | 49101 | 146940 | 9.1 |
| 11 - 12 | 148240 | 52243 | 149395 | 9.6 |
| 12 - 13 | 206444 | 67394 | 179200 | 9.5 |
| 13 - 14 | 185175 | 61513 | 164897 | 8.4 |
| 14 - 15 | 351491 | 103789 | 163520 | 12.5 |
| 15 - 16 | 978155 | 165299 | 234832 | 32.5 |
| 16 - 17 | 835257 | 158711 | 231822 | 33.6 |
| 17 - 18 | 745095 | 154450 | 228337 | 32.8 |
| 18 - 19 | 722444 | 153898 | 228000 | 37.2 |
| 19 - 20 | 644618 | 146167 | 221226 | 32.2 |
| 20 - 21 | 621817 | 142464 | 225409 | 30.0 |
| 21 - 22 | 634095 | 143889 | 230248 | 31.1 |
| 22 - 23 | 642385 | 142940 | 233102 | 30.2 |
| 23 - 24 | 629936 | 138903 | 229783 | 29.5 |

### C. Topic Extraction Using High Performance Singular Vector Decomposition

This section introduces the experimental result of the first method [4].

*1) Step 2: Clustering:* Then TF-IDF for $(a_{ij})_t$ are calculated and $redsvd$ [8] with rank = 10 has been applied. The performance of redsvd was reasonable. For example, the run-time of redsvd for the matrix during 15:00-16:00 on March 11 (165299 authors $\times$ 234832 words) was less than 10 seconds. We formed clusters by $k$-means by setting k = 5. From Figure 3, we realize that authors could be divided into five clusters.

*2) Step 3: Feature Selection:* For five clusters of each time slot, CWC has been adopted for feature selection. *Matthew's Correlation Coefficient* (MCC) [19] is used to order extracted feature words whose score ranges from $-1$ to 1. The words with high MCC value ($> 0$) positively express the feature of the cluster while the words with low MCC value ($< 0$) negatively express the feature of the cluster they belong to. To extract feature words for representation of each cluster, positive words are selected. (All

TABLE III
FEATURE SELECTION RESULT DURING 15:00-18:00 ON MAR. 11

| Time Slot | Cluster # | # of Authors | # of Words | CWC Runtime ($msec$) | # of Feature words | Excerpts from Feature words ( ) shows MCC value |
|---|---|---|---|---|---|---|
| 15:00-16:00 | 0 | 40354 | 38822 | 72298 | 254 | Earthquake(0.2940) all right(0.2277) message(−0.1610) use(−0.1438) use(−0.1312) disaster(−0.1415) net(−0.1250) aftershock(0.1438) Twitter(−0.1082) hope(−0.1094) so(0.1113) worry(0.1046) tsunami(0.0981) kana(0.0876) need(−0.0794) please(−0.0850) confirmation(−0.0896) diffusion(−0.0888) Mr.(0.0868) seismic intensity(0.0892) successfully(0.0845) Tokyo(0.0761) shaking(0.0753) Tohoku(0.0653) |
| | 1 | 7956 | 38822 | 55080 | 42 | Emergency(0.4564) net(0.4518) use(0.4396) ask(0.4356) Bath(0.4239) tsunami warning(0.4138) location(0.3688) telephone(0.3561) RT(0.3555) evacuation(0.3645) absolute(0.3354) everyone(0.3465) possible(0.3178) information(0.3382) so(0.3425) preparation(0.3114) Miyagi(0.3324) possibility(0.2983) it(0.3193) Great Hanshin Earthquake(0.2889) contact(0.3067) |
| | 2 | 89182 | 38822 | 63135 | 227 | Telephone(−0.5044) use(−0.4263) diffusion(−0.4626) disaster(−0.4458) confirmation(−0.4625) safety(−0.4434) hope(−0.4325) earthquake(−0.4915) message(−0.4128) Bathing(−0.3819) net(−0.3850) experience(−0.3799) please(−0.3878) Tsuitta(−0.3916) rice(−0.3596) Great Hanshin-Awaji Earthquake(−0.3533) electricity(−0.3608) |
| | 3 | 14466 | 38822 | 62668 | 85 | Telephone(0.2888) tsunami warning(0.2729) experience(0.2626) confirmation(0.2710) evacuation(0.2688) contact(0.2484) diffusion(0.2472) information(0.2332) earthquake(0.2367) Fire(0.2073) electricity(0.2197) disaster(0.2234) Miyagi(0.2255) tsunami(0.2231) location(0.2033) safety(0.2022) |
| | 4 | 9614 | 38822 | 39734 | 66 | Dial(0.5629) use(0.5281) message(0.5149) emergency(0.5147) Twitter(0.5040) safety(0.4831) net(0.4787) use(0.4643) hope(0.4402) diffusion(0.4162) ask(0.3341) Fukushima Prefecture(0.1608) magnitude 5(0.1559) earthquake(−0.1769) all right(−0.1375) earthquake information(0.1073) aftershocks(−0.1151) |
| 16:00-17:00 | 0 | 103114 | 37659 | 76145 | 263 | Diffusion(−0.6629) hope(−0.6393) Asakusa(−0.4423) Tokyo(−0.4577) so(−0.4663) power failure(−0.4476) tsunami warning(−0.4131) earthquake(−0.4680) confirmation(−0.4393) net(−0.3967) evacuation(−0.4307) Miyagi(−0.4035) it(−0.4291) information(−0.4093) |
| | 1 | 8823 | 37659 | 47497 | 40 | Hope(0.3876) refuge(0.3885) big tsunami alert(0.3621) outage(0.3606) confirmation(0.3587) hill(0.3449) possibility(0.3438) case(0.3410) BLEMMER(0.3385) Miyagi(0.3391) telephone(0.3293) Intelligence(0.3266) yuan bolt(0.2976) drink water(0.2949) Yun Yan(0.3142) Jin wave(0.3158) may(0.2890) Note(0.2989) earthquake(0.2988) coast(0.2825) |
| | 2 | 9629 | 37659 | 55416 | 48 | Asakusa(0.8184) Gikuhausu(0.8145) Tokyo(0.5900) Who(0.4552) real(0.4023) Search(0.3931) abdomen(0.3840) mackerel(0.3783) hoax(0.3445) important(0.2679) Twitter(0.2565) diffusion(0.2732) location(0.2227) emergency(0.1813) net(0.2001) information(0.1783) |
| | 3 | 1214 | 37659 | 60294 | 34 | Bleeding(0.2760) hemostasis(0.2352) drinking water(0.2456) the main cock(0.2430) roar(0.2089) possible(0.2395) rescue(0.2361) woman(0.2170) Konkurito(0.2226) leakage Bureka(0.2220) advice(0.2260) moment(0.2141) mobile phone(0.2281) . ※ (0.1912) if(0.2413) police(0.2203) Hanshin(0.2108) Supido(0.2128) |
| | 4 | 32613 | 37659 | 56202 | 111 | Diffusion(0.4344) hope(0.3969) earthquake(0.2947) power failure(0.2791) confirmation(0.2699) it(0.2727) so(0.2657) telephone(0.2575) Large tsunami warning(0.2362) evacuation(0.2459) Miyagi(0.2354) tsunami(0.2423) safety(0.2160) disaster(0.2054) like(0.2089) message(0.2019) |
| 17:00-18:00 | 0 | 17613 | 37601 | 45228 | 82 | Diffusion(0.3575) hope(0.2898) earthquake(0.2680) so(0.2584) maximum(0.2312) ask(0.2281) evacuation(0.2255) shaking(0.2058) time(0.2017) disaster(0.1941) Great Hanshin-Awaji Earthquake(0.1881) it(0.1903) information(0.1859) Free(0.1755) so(0.1786) telephone(0.1744) for(0.1708) |
| | 1 | 83658 | 37601 | 54149 | 218 | Diffusion(−0.5298) earthquake(−0.5160) hope(−0.4457) so(−0.4230) evacuation(−0.3773) please(−0.3654) maximum(−0.3591) disaster(−0.3351) it(−0.3494) because(−0.3209) information(−0.3284) Note(−0.3040) contact(−0.3261) shaking(−0.3180) tsunami(−0.3209) so(−0.3250) |
| | 2 | 38796 | 37601 | 67161 | 234 | Earthquake(0.02471) diffusion(0.01092) contact(0.00663) so(0.00534) hope(0.00485) family(0.00456) it(0.00447) maximum(0.00398) so(0.00389) all right(0.003710) today(0.003611) successfully(0.003512) aftershock(0.003313) worry(0.003314) because(0.002816) after(0.002617) tsunami(0.002519) provides(0.002520) ask(0.002521) shaking(0.002322) like(0.002123) time(0.002024) confirmation(0.002025) information(0.001926) |
| | 3 | 8035 | 37601 | 56479 | 28 | Diffusion(0.3407) evacuation(0.3465) hope(0.3371) disaster(0.3206) so(0.3208) Note(0.2976) shelter(0.2794) ask(0.2896) absolute(0.2669) blankets(0.2592) information(0.2818) the vicinity(0.2640) current(0.2611) risk(0.2511) telephone(0.2720) earthquake(0.2707) location(0.2558) prepared(0.2425) tsunami(0.2656) it(0.2616) confirmation(0.2537) Great Hanshin Earthquake(0.2312) |
| | 4 | 2581 | 37601 | 28272 | 34 | Woman(0.3668) risk(0.3490) absolute(0.3500) shelter(0.3505) crime(0.3340) Note(0.3491) disaster(0.3483) open(0.3408) current(0.3373) If(0.3301) everyone(0.3348) possibility(0.3300) location(0.3287) rescue(0.3065) evacuation(0.3177) use (0.3048) Hanshin Earthquake (0.3151) emergency (0.3138) possible(0.2956) |

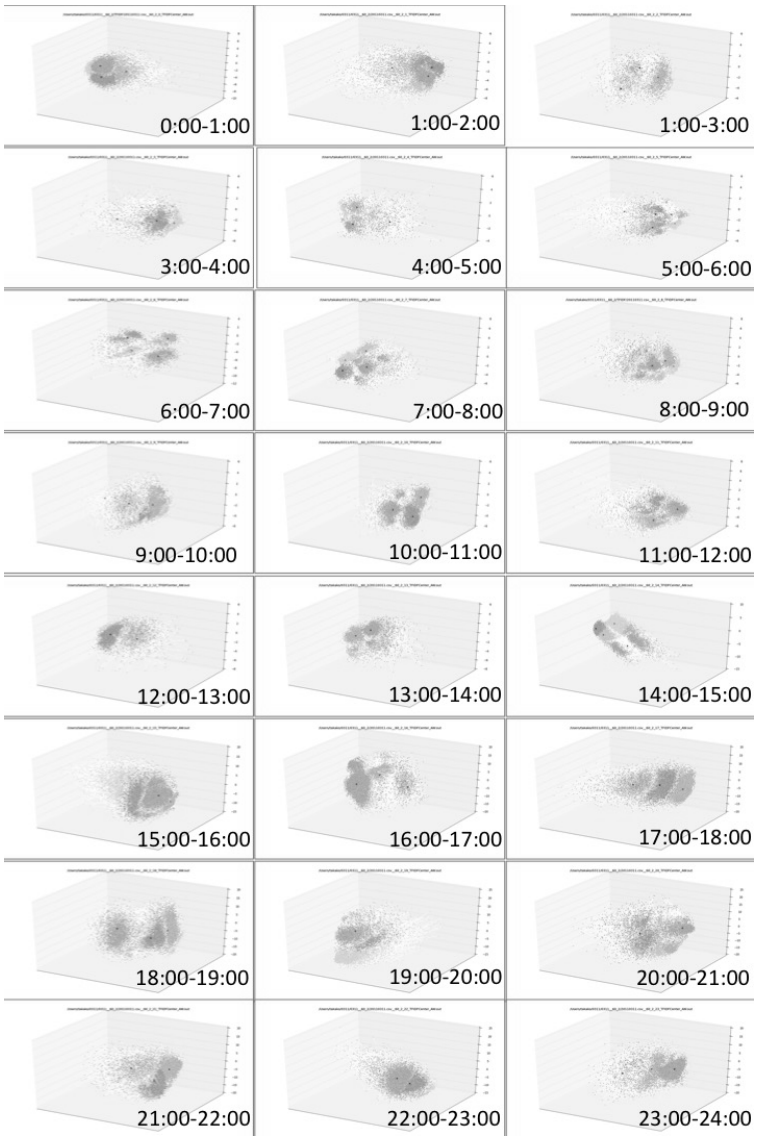


Fig. 4. Clustering Results by SVD and $k$-means during 0:00-24:00 on Mar. 11 (by hour)

words were originally in Japanese, but translated to English.)

Table III shows the feature selection result during 15:00-18:00 on Mar. 11. According to the feature words in Table III, the topic of each cluster is observed as follows:

- March 11 15:00-16:00
  - $cluster0$: Damage after the quake

  - $cluster1$: Emergency call on the quake
  - $cluster2$: No specific topic
  - $cluster3$: Tsunami warning and evacuation
  - $cluster4$: Message dial for the quake for confirming safety
- March 11 16:00-17:00
  - $cluster0$: No specific topic
  - $cluster1$: Escape from Tsunami with hope
  - $cluster2$: Hoax on Twitter/net
  - $cluster3$: Injury due to the quake
  - $cluster4$: Diffusion of hope, power failure
- March 11 17:00-18:00
  - $cluster0$: Diffusion of hope
  - $cluster1$: No specific topic
  - $cluster2$: Diffusion of damaged situation
  - $cluster3$: Diffusion of evacuation situation
  - $cluster4$: Risk of women after the quake

Extracted feature words with positive MCC in the $cluster0$ during15:00-16:00 on March 11 were "Earthquake", "all right", "aftershock", "so", "worry" and son. These words can be interpreted as "after the earthquake, people were worried about the damage of the quake". For the $cluster1$ during 15:00-16:00 on March 11, extracted feature words with positive MCC were "Emergency", "net", "use", "ask", "tsunami warning", "location", "telephone" and so on. This may show that people used the emergency call after the quake. On the other hand, for the $cluster3$ during 15:00-16:00 on March 11, extracted feature words with positive MCC were "Telephone", "tsunami warning", "experience", "confirmation", "evacuation", "contact" and so on. The $cluster4$ also had "Dial", "use", "message", "emergency", "Twitter," "safety", "net2","use", "hope", "diffusion", "ask" and so on as extracted feature words with positive MCC. This is also estimated that people used a message dial for confirming safety. However, feature words of $cluster3$ and $cluster4$ are similar with $cluster2$. They can be considered as the same cluster.

Of course, the $cluster4$ in 17:00-18:00 on March 11 showed the topic about the risk of women after the quake, some clusters showed their topics relatively clearly. As the number of clusters are set in advance, the clustering results did not seem to work well in most of the cases.

### *D. Topic Extraction Using High Performance Feature Selection Technique*

This section introduces the experimental result of the second method [5].

*1) Step 2: Feature Selection Technique Adaptation:* Next, we applied CWC to extract principal word vectors from these matrices. We set the 9:00-10:00 matrix as the initial matrix, because in the period of 9:00-10:00 on March 11, the earthquake did not happen yet so that it can be an ordinary state. And we created an input file with two word presence vectors for each author, one for the tweets that were sent from 9:00-10:00 on March 11, and one for tweets that were sent between 10:00 on March 11 and 24:00 on March 12. Tweets that were sent during the 9:00-10:00 hour were placed in class 0, and tweets that were sent at other times were placed in class 1. Figure 5 shows an example of an input file for CWC. It contains a word list presented in both classes, a class label, and the author-words matrix for each class.

We then created the same type of file for all subsequent hours. CWC could then extract a principal word vector for each target hour $t$. Table IV shows the number of extracted principal word vectors, some principal word vector examples derived by CWC, and their runtimes. For example, CWC extracted 4452 feature word vectors in 525485 ms from the March 11, 15:00-16:00 matrix. Most run-times were around 300,000-600,000 ms. Considering the size of each matrix, the performance is reasonable enough.

TABLE IV
PRINCIPAL WORD VECTORS EXTRACTED FOR MARCH 11 BY CWC AND THEIR RUN TIME

| Time Slot (24h) | # of Principal Words Vector | CWC Runtime (msec) | Excerpts from Principal Word Vector |
|---|---|---|---|
| 10:00-11:00 | 6153 | 238742 | Application0.0538 Noon0.0546 Present0.0509 Complete0.0451 Plan0.0437 Lunch0.0335 |
| 11 00-12 00 | 6393 | 315264 | Realize0.0062 F K:0.0012 Korea0.0068 Sushi0.0042 HarusameNodle0.005 Jiten0.0036 |
| 12:00-13:00 | 7132 | 315264 | Noon0.0908 Iitomo0.0574 LunchBreak0.0544 Lunch-0.0500 LunchBox0.0456 Rice0.0452 |
| 13:00-14:00 | 6876 | 292992 | SakuraShinjuu0.0546 Noon0.0577 DoCoMo0.0537 Quittance0.0474 Muka0.0466 Marron0.0450 |
| 14:00-15:00 | 6002 | 405205 | Earthquake0.4153 Seismic intensity0.2300 Miyagi-0.2128 All right0.2163 Shaking0.1582 |
| 15:00-16:00 | 4452 | 525485 | Earthquake0.3760 All right0.3064 Telephone0.2691 Safe0.2380 Aftershocks0.2226 Evacuation0.2206 Miyagi0.2170 Tsunami0.2132 Confirmation0.2204 Disaster0.2022 Diffusion0.2173 Safety0.1991 Seismic intensity0.2006 Worry0.2020 Message0.1779 |
| 16:00-17:00 | 4406 | 497594 | Earthquake0.3111 diffusion0.2689 Good morning-0.2747 Safe0.2331 Evacuation0.2147 Aftershocks0.2011 Tsunami0.1969 Hope0.2087 Shaking0.1865 all right0.2131 contact0.1947 maximum0.1851 Disaster0.1683 Power failure0.1665 |
| 17:00-18:00 | 4394 | 533004 | Earthquake0.3166 Diffusion0.2730 tsunami0.2457 Safe-0.2481 Hope0.2515 Evacuation0.2259 All right0.2424 Power outage0.2061 Telephone0.2198 Confirmation-0.2037 Aftershocks0.1840 Contact0.1957 Miyagi0.1769 Information0.1922 Worry0.1826 |
| 18:00-19:00 | 5237 | 599049 | Earthquake0.2667 Diffusion0.2612 Evacuation0.2354 Open0.2168 Sae0.2221 Hope0.2149 Place.2093 Disaster0.1935 Damage0.1804 Free0.1893 Go home0.1802 All right0.2000 Information0.1923 Ribatitawa0.1623 Aftershocks0.1617 Power failure0.1604 |
| 19:00-20:00 | 4650 | 460319 | Earthquake0.2599 Evacuation0.2499 Diffusion0.2526 Open0.2288 Location0.2400 Safe0.2265 Hope0.2117 Home0.2007 Information0.1990 Free0.1876 Disaster-0.1643 |
| 20:00-21:00 | 5317 | 514139 | Earthquake0.2514 Diffusion0.2563 Safe0.2352 Evacuation0.2080 Information0.2254 Open0.2010 Hope0.2189 Home0.1897 Location0.1926 Aftershocks0.1710 Resume0.1670 Contact0.1757 All right0.1916 Tsunami0.1561 Ask0.1799 Power failure0.1515 Disaster0.1437 Worry0.1576 Shelter0.1401 |
| 21:00-22:00 | 4700 | 448331 | Earthquake0.2573 Diffusion0.2604 Safe0.2370 Evacuation0.2184 Hope0.2321 Information0.2167 Resume0.1822 Give me0.2030 Power failure0.1632 Operation0.1709 Contact0.1749 Open0.1613 Aftershocks0.1581 Go home0.1628 All right0.1822 Recovery0.1528 Tsunami0.1484 |
| 22:00-23:00 | 4794 | 478048 | Earthquake0.2628 Safe0.2430 Diffusion0.2448 Hope-0.2112 Evacuation0.1881 Tsunami0.1852 Information-0.2076 Aftershocks0.1723 Ask0.1897 Worry0.1770 Damage0.1589 |
| 23:00-24:00 | 4554 | 400798 | Earthquake0.0531 successfully0.0483 Diffusion0.0475 Hope0.0356 Tsunami0.0355 Evacuation0.0345 Information0.0323 Aftershocks0.0312 Worry0.0301 Ask0.0287 Contact0.0265 |

TABLE V
MANHATTAN DISTANCE CALCULATION RESULT

| Time Slot (Date Hour) | Manhattan Distance | Time Slot (Date-Hour) | Manhattan Distance |
|---|---|---|---|
| 03/11/2011 10-11 | INITIAL | 03/12/2011 5-6 | 0.89 |
| 03/11/2011 11-12 | 0.62 | 03/12/2011 6-7 | 0.89 |
| 03/11/2011 12-13 | 0.65 | 03/12/2011 7-8 | 0.88 |
| 03/11/2011 13-14 | 0.66 | 03/12/2011 8-9 | 0.88 |
| 03/11/2011 14-15 | 0.80 | 03/12/2011 9-10 | 0.88 |
| 03/11/2011 15-16 | 0.88 | 03/12/2011 10-11 | 0.88 |
| 03/11/2011 16-17 | 0.89 | 03/12/2011 11-12 | 0.93 |
| 03/11/2011 17-18 | 0.89 | 03/12/2011 12-13 | 0.88 |
| 03/11/2011 18-19 | 0.88 | 03/12/2011 13-14 | 0.88 |
| 03/11/2011 19-20 | 0.88 | 03/12/2011 14-15 | 0.88 |
| 03/11/2011 20-21 | 0.87 | 03/12/2011 15-16 | 0.87 |
| 03/11/2011 21-22 | 0.88 | 03/12/2011 16-17 | 0.86 |
| 03/11/2011 22-23 | 0.88 | 03/12/2011 17-18 | 0.88 |
| 03/11/2011 23-24 | 0.88 | 03/12/2011 18-19 | 0.87 |
| 03/12/2011 0-1 | 0.88 | 03/12/2011 19-20 | 0.86 |
| 03/12/2011 1-2 | 0.88 | 03/12/2011 20-21 | 0.89 |
| 03/12/2011 2-3 | 0.88 | 03/12/2011 21-22 | 0.86 |
| 03/12/2011 3-4 | 0.89 | 03/12/2011 22-23 | 0.86 |
| 03/12/2011 4-5 | 0.89 | 03/12/2011 23-24 | 0.86 |

TABLE VI
AUTHOR-WORD MATRICES ON MAR. 11

| ID | Time Slot (Date Hour) | Principal Word Vector (excerpts) |
|---|---|---|
| A | Mar. 11 18:00-19:00 | Open(0.3142), Shinagawa Prince Hotel(0.2349), Building(0.2275), Meiji(0.2147), Hamamatsu Station(0.2130), Hamamatsu(0.2127), Ikebukuro(0.2261), Tamachi(0.2126), Shinagawa(0.2201), Naka-ku(2118), shelter(0.2216), Shinjuku(0.2308), tea(0.2139), |
| B | Mar. 12 4:00-5:00 | Nagano(0.4221), Niigata(0.3618), Earthquake Early Warning(0.3096), Nagano Prefecture(0.2059), this time(0.2171), Japan(0.2020), Chuetsu(0.1639), morning(0.1420), Chuetsu region(0.1364), |
| C | Mar. 12 6:00-7:00 | Earthquake Emergency Warning(0.2819), Good morning(0.2459), Nagano(0.1886), Kanagawa(0.1842), radioactivity(0.1532), rescue(0.1624), nuclear power plant(0.1626), Fukushima Daiichi nuclear power plant(0.1461), radius(0.1462), |
| D | Mar. 12 10:00-11:00 | Power-saving(0.2506), power(0.2264), shortage(0.2009),support(0.1860), rescue(0.1761), Good morning(0.1631), name(0.1658), donations(0.1570), yesterday(0.1719), today(0.1671), |
| E | Mar. 12 21:00-22:00 | Power-saving(0.2803), donations(0.2077), nuclear power plant(0.2180), Yashima strategy(0.1690), power(0.1721), home(0.1631), reactor(0.1474), disaster land(0.1757), explosion(0.1577), |

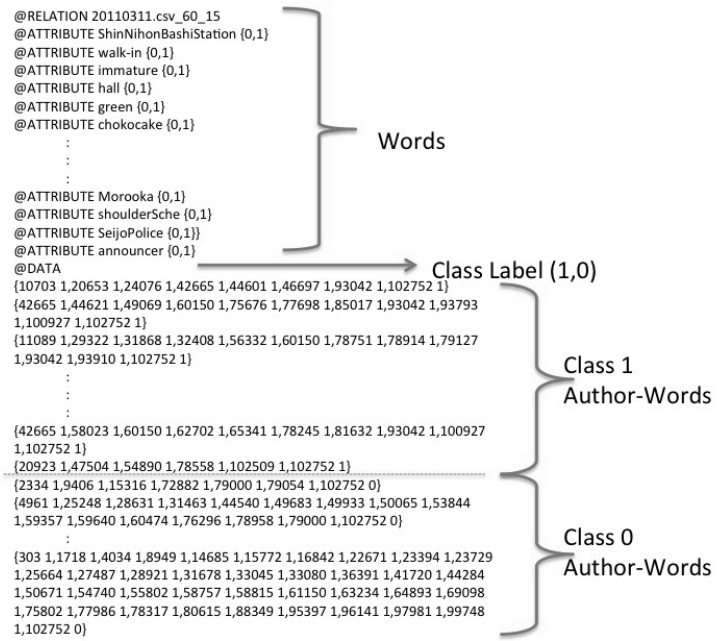


Fig. 5. Example of CWC input

The table also shows that after the earthquake (15:00), the principal word vectors were made of mostly earthquake-related words (e.g. ”earthquake,” ”safe,” ”aftershocks,” and ”seismic Intensity”). Since conventional methods such as LDA and LSA require an unreasonable amount of memory to process a dataset of this size, it is not easy to compare our proposed method with these earlier ones.

*2) Step 3: Distance Calculation:* Then we calculated the Manhattan Distance using MCC values that were computed by CWC. Table V shows the Manhattan Distance of each hour's principal word vector from the initial principal word vector, and Figure 6 shows changes in Manhattan Distance over time.

*3) Step 4: Event Detection:* Obviously, after the Earthquake (after 15:00 on March 11), the distance increased. This change suggests that a significant event happened at that time. However, after 15:00 on March, the distance from the 15:00 principal word vector remains at a high level. We know, however, that the Great East Japan Earthquake was really a large event made up of many smaller events: the earthquake, a tsunami that resulted from the earthquake, the evacuation of the area around the Fukushima Daiichi Nuclear Plant. we know other significant events happened that were related to the earthquake around this time. We would like to further break down the super-event of the Great East Japan Earthquake to discover the related sub-events within the super-event. Therefore we have to break down the events after 15:00 on March 11.

*4) Iteration of Step 2 - 3, and Step 4:* To detect nested events, we repeated *Steps 2-4*. We set the matrix of March 15:00-16:00 as the initial Matrix $A_0$. We went back to the Step 2 and derived the principal word vectors for each following hour. We chose the principal word vector of March 11 16:00-17:00 as the initial principal word vector, and computed the distance between each hour's principal word vector and the initial principal word vector again. Figure 7 shows the resulting Manhattan Distance calculations.

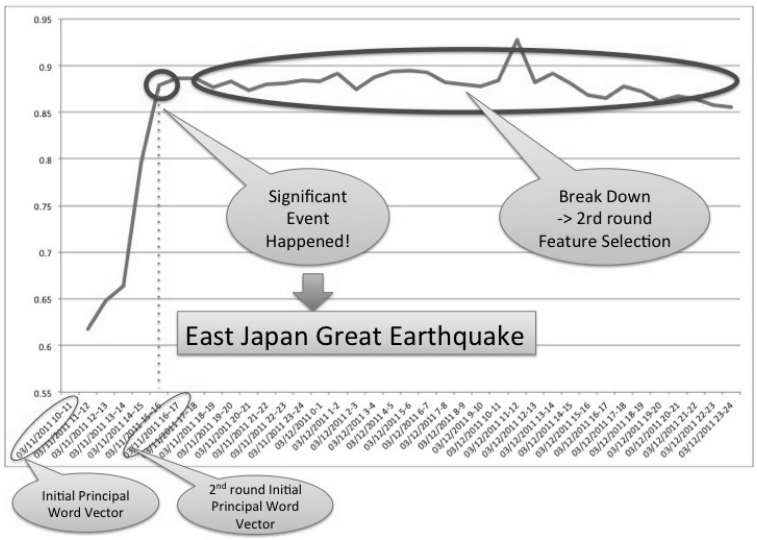


Fig. 6. Manhattan Distance during 10:00-23:00 on Mar. 11 and 0:00-23:00 on Mar.12

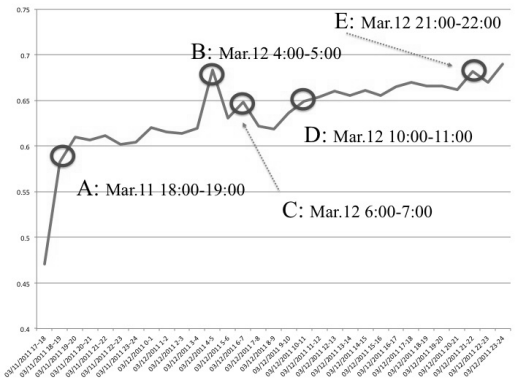


Fig. 7. Manhattan Distance from 17:00-23:00 of Mar. 11 to 24:00 and 0:00-23:00 of Mar.12

In Figure 7, we found five large distance changes *A, B, C, D, E* that are circled. Each of these changes must have a specific word vector that can identify the events that caused the changes. Table VI shows the principal word vector in each change.

At *A* (during 18:00-19:00 on March 11), the principal words were ”open”, ”Sinagawa Prince Hotel”, ”Hamamatsu Station”, ”Tamachi” and so on. While these may seem like strange words to use in the immediate aftermath of a large earthquake, they reflect twitter users' immediate concerns: most train stations were closed, stranding many people at work or school. At around 18:00, however, some stations re-opened and commuters were able to go home. At 4:00 AM on March 12, at *B*, a severe aftershock with

magnitude 5.9 struck Nagano Prefecture, Japan; as a result, the principal words for this hour included "Nagano", "earthquake", and "Niigata". Later that day, at *C* (during 6:00-7:00 on March 12), concern grew over damage to the Fukushima Daiichi Nuclear Plant, causing the principal words to be "radioactivity", "nuclear power plant", "Fukushima Daiichi Nuclear Plant", and "radius", in addition to "Nagano" and "Earthquake Emergency Warning". At *D* (during 10:00-11:00 on March 12), we see the growth of two related problems: blackouts resulting from damage to the power infrastructure, and relief efforts for those affected by the earthquake. The resulting principal words were "power-saving", "power", "shortage", "support", "donation", and "relief goods". Finally, at *E* (during 21:00-22:00 on March 12), we see principal words "Yashima strategy" in addition to "power-saving", "power" and so on. "Yashima strategy" was a name spontaneously socialized in twitter regarding power-saving after the earthquake.

Extracting these five sub-events through iterating *Step 2, Step 3 and Step 4* allows us to analyze events more precisely than making a single pass.

## V. Discussion: Issues on Conventional Social Media Analysis Method

As we described in Section II, generally, the conventional social media analysis method has a scalability problem. Existing data mining technique target thousands of items, not millions. In addition to lack of scalability, we believe there are several problems.

First, the accuracy of clustering (decomposition) techniques is not high, nor can these techniques deliver reasonable performance. Most of the clustering techniques like $k$-means require the number of clusters to be estimated in advance which lowers cluster quality.

Next, to extract important keywords from clusters, word scoring methods such as TF-IDF [14] or term-score [20] are generally used. However, such scoring methods are based on word occurrence, and high-frequency words tend to be extracted. Therefore, word scoring methods cannot always represent each cluster with high precision.

Third, in this paper, the original technique CWC for feature selection has been utilized, yet even using CWC, it is not easy to extract appropriate words from low quality clusters.

Finally, sometimes these methods identify false similarities between clusters over time.

To overcome these issues, development of new method for social media analysis is required.

## VI. Conclusion

This paper introduced an improvement of the conventional word vector-based topic detection method for social media by using high performance Singular Vector Decomposition library $redsvd$ and k-means to identify topic clusters over time from the huge data set of over two hundred million tweets related to the Great East Japan Earthquake. The fast feature selection technique CWC has also been utilized to extract features from each cluster. The proposed technique confirmed the feasibility of topic extraction from big data. From the experiment, though the emergent topics can be observed from the authors' clusters, the issues of conventional topic detection techniques from big data can also be identified as well. To overcome the issues on social media analysis, we plan to develop new social media analysis method that can achieve better performance and accuracy.

In this paper, we proposed the event detection method for big data using our own fast feature selection CWC. CWC allowed us to identify events with great speed and accuracy in over 200 million tweets from the Great East Japan Earthquake. For our future work, we intend to apply our methods to other data, and to develop a process for detecting the bursts automatically.

## Acknowledgment

The author would like to thank the Chiba University of Commerce for giving the opportunity to stay at UCLA as the sabbatical leave. This paper was supported by the Grant-in-Aid for Scientific Research (KAKENHI Grant Numbers 26280090, and 15K00314) from the Japan Society for the Promotion of Science.

## REFERENCES


[1] S. T. Dumais , *A Graph Analytical Approach for Topic Detection*, Annual Review of Information Science and Technology, 38: 188, doi:10.1002/aris.1440380105, 2005.

[2] H. Sayyadi and L. Raschid, *Latent Semantic Analysis*, ACM Transactions on Internet Technology ,13(2), Article No. 4, November 2013.

[3] D. M. Blei, A. Y. Ng and M. I. Jordan, *Latent Dirichlet Allocation*, Journal of Machine Learning Research, 3 (4-5), pp. 993-1022, doi:10.1162/jmlr.2003.3.4-5.993, 2003.

[4] T. Hashimoto, T. Kuboyama and B. Charkaborty, *Topic Extraction from Millions of Tweets using Singular Value Decomposition and Feature Selection*, Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) 2015, 2015.

[5] T. Hashimoto, D. Shepard, T. Kuboyama, and K. Shin, *Event Detection from Millions of Tweets Related to the Great East Japan Earthquake Using Feature Selection Technique*, 2015 IEEE 15th International Conference on Data Mining Workshops (ICDMW 2015), pp.7-12, 2015

[6] K. Shin, T. Kuboyama, T. Hashimoto, and D. Shepard, *Super-CWC and super-LCC: Super fast feature selection algorithms*, Proc. 2015 IEEE International Conference on Big Data (Big Data), pp. 1-7, 2015.

[7] T. Hashimoto, T. Kuboyama and Y. Shirota *Topic Detection about the East Japan Great Earthquake based on Emerging Modularity*, Volume 251: Information Modelling and Knowledge Bases XXIV, pp. 110-126, 2013.

[8] redsvd, https://code.google.com/p/redsvd/.

[9] J. B. MacQueen, *Some Methods for classification and Analysis of Multivariate Observations*, Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press. pp. 281-297, 1967.

[10] I. Fujino and Y. Hoshino, *A Method for Identifying Topics in Twitter and its Application for Analyzing the Transition of Topics*, Proc. DEIM Forum 2014, C4-2, 2014.

[11] M. J. Paul and M. Dredze, *Discovering Health Topics in Social Media Using Topic Models*, PLoS ONE 9(8): e103408. doi:10.1371/journal.pone.0103408, 2014.

[12] W. X. Zhao, J. Jiang, J. Weng, J. He, E. P. Lim, H. Yan and X. Li, *Comparing Twitter and Traditional Media Using Topic Models*, Proc. the 33rd European Conference on Information Retrieval(ECIR 2011), LNCS 6611, pp. 338-349, 2011.

[13] T. Kitada, K. Kazama, T. Sakaki F. Toriumi, A. Kurihara, K. Shinoda, I. Noda and K. Saito, *Analysis and Visualization of Topic Series Using Tweets in Great East Japan Earthquake*, The 29th Annual Conference of the Japanese Society for Artificial Intelligence, 2B3-NFC-02a-1, 2015.

[14] H. C. Wu, R. W. P. Luk, K. F. Wong and K. L., Kwok, *Interpreting TF-IDF term weights as making relevance decisions*, ACM Transactions on Information Systems, 26 (3), doi:10.1145/1361684.1361686, 2008.

[15] Z. Zhao and H. Liu. Searching for interacting features. *In Proceedings of International Joint Conference on Artificial Intelligence*, pp. 1156-1161, 2007.

[16] Hottolink, Inc., http://www.hottolink.co.jp/english.

[17] J. Kleinberg, *Bursty and Hierarchical Structure in Streams*, Proc. 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 2002.

[18] MeCab: Yet Another Part-of-Speech and Morphological Analyzer, http://taku910.github.io/mecab/

[19] B. W. Matthews, *Comparison of the predicted and observed secondary structure of T4 phage lysozyme*, Biochimica et Biophysica Acta (BBA) - Protein Structure, 405 (2), pp.442-451, 1975.

[20] D. M.Blei and J. D. Lafferty, *Text Mining: Theory and Applications*, chapter TOPIC MODELS, Taylor and Francis, 2009.

[21] P. E. Black, *Manhattan distance*, in Dictionary of Algorithms and Data Structures [online], Vreda Pieterse and Paul E. Black, eds. 31 May, 2006.

[22] Y. Zhu, and S. Dennis Shasha, *Efficient Elastic Burst Detection in Data Streams*, Proc. SIGKDD ’03, Washington, DC, USA. 2003.


（2016.1.20 受稿，2016.3.8 受理）