An Integrated Data Model for Management and Mining of Social Big Data: Concepts and Applications

Hiroshi Ishikawa
Faculty of System Design
Tokyo Metropolitan University
Tokyo, Japan
e-mail: ishikawa-hiroshi@tmu.ac.jp

Richard Chbeir
LIUPPA Lab.
University of Pau and Adour Countries
Anglet, France
e-mail: richard.chbeir@univ-pau.fr

***Abstract*— Big data typically include three kinds of data such as IoT (Internet of Things) data, social data, and open data. However, there exists no integrated analytic methodology as to big data applications involving different kinds of data sources. Furthermore, big data applications inherently involve data management and data mining. We propose an integrated analytic methodology by taking a data model approach. We propose an abstract data model for seamlessly describing both data management and data mining cooccurring in big data applications, based on mathematical concepts of families, that is, collections of sets. Our model also facilitates reproducibility and accountability required for promoting social big data researches and developments. We have validated our proposal by adapting our model to a variety of real case studies in social big data.**

Keywords-social data; big data; data model; data management; data mining; tourism; disaster prevention.

I. INTRODUCTION

In the present age, large amounts of data are produced continuously in science, on the Internet, and in physical systems. Such phenomena are collectively called data deluge. And needs for analytic methodologies dedicated to data deluge have emerged [1].

According to some researches carried by International Data Corporation, or IDC [2][3] for short, the size of data which are generated and reproduced all over the world every year is estimated to be 161 Exabytes as of the published year. The total amount of data produced in 2011 exceeded 10 or more times the storage capacity of the storage media available in that year. The size is forecasted to grow to 40, 000 Exabytes by 2020. Experts in scientific and engineering fields produce a large amount of data by observing and analyzing the target phenomena. Even ordinary people voluntarily post a vast amount of data via various social media on the Internet. Furthermore, people unconsciously produce data via various actions detected by physical systems, such as sensors and Global Positioning System, or GPS for short, in the real world. It is expected that such data can generate various values. In the above-mentioned research report of IDC, data produced in science, the Internet, and in physical systems are collectively called big data. The features of big data can be summarized as follows:

- The quantity (Volume) of data is extraordinary, as the name denotes.
- The kinds (Variety) of data have expanded into unstructured texts, semi-structured data, such as XML, and graphs (i.e., networks).
- As is often the case with Twitter and sensor data streams, the speed (Velocity) at which data are generated is very high.

Therefore, big data is often characterized as V^3 by taking the initial letters of these three terms Volume, Variety, and Velocity. Big data are expected to create not only knowledge in science but also derive values in various commercial ventures. "Volume" and "velocity" require more computing power than ever before. "Variety" implies that big data appear in a wide variety of applications and then data have a wide variety of structures.

Further, big data inherently contain "vagueness," such as inconsistency and deficiency. Such vagueness must be resolved in order to obtain quality analysis results. Moreover, a recent survey done in Japan has made it clear that a lot of users have other "vague" concerns as to the securities and mechanisms of big data applications [4]. In other words, service providers deploying big data have accountability for explaining to generic users among stake holders how relevant big data are used. The resolution of such concerns is one of the keys to successful diffusion of big data applications. In the sense that "vagueness" is a fourth V, V^4 should be used to characterize big data, instead of V^3.

Big data typically include three kinds of data: *IoT (Internet of Things) data* collected by a variety of networked sensors and mobile gadgets in real physical worlds, *social data* posted at social media sites, such as Twitter and Flickr, and *open data* published for everyone to access.

To our knowledge, however, there exists no analytic methodology as to applications involving different sources of big data. Furthermore, such big data applications are often mixed of data management and data mining from our observations. This integration is also needed for describing such applications.

Section II introduces concepts of social big data as an integrated analysis methodology to big data and describes rationales for a data model approach to it. Section III introduces our data model for social big data. Section IV explains case studies by adapting the model to them in order to verify the validity of the model.

II. RATIONALES FOR DATA MODEL

A. Social Big Data

In big data applications, especially, cases where two or more data sources including at least one social data source are involved, are more interesting from a viewpoint of

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

usefulness to businesses [4]. If more than one data source can be analyzed by relating them to each other, and by paying attention to the interactions between them, it may be possible to understand what cannot be understood, by analysis of only either of them. For example, even if only sales data are deeply analyzed, reasons for a sudden increase in sales, that is, what has made customers purchase more products suddenly, cannot be known. By analysis of only social data, it is impossible to know how much they contributed to sales, if any. However, if both sales data and social data can be analyzed by relating them to each other, it is possible to discover why items have begun to sell suddenly, and to predict how much they will sell in the future, based on the results. In a word, such integrated analysis is expected to produce bigger values than otherwise. While big data simply imply data with V^3 or V^4 features, we would like to call such an integrated analytic methodology with respect to big data *Social Big Data*, or *SBD* for short. Throughout this paper, we use SBD instead of social big data.

Even if only one social data source, such as Twitter articles and Flickr images is available and if such articles and images have geo-tags (i.e., location information), as well, social big data mining is useful. That is, by collecting those articles and images based on conditions specified with respect to locations and time intervals and counting them for each grid (i.e., unit location), probabilities that users post such data at the locations can be basically computed. By using such probabilities, human activities can be analyzed, such as probabilities of foreigners staying at specific spots or those moving from one spot to another. The results will be applied to tourism and marketing.

Of course, locations and time can also be used in joining different sources of data in SBD applications. Further, a certain level of location can be represented as a collection of lower levels of locations. Similarly, a time interval can be divided into a collection of shorter time intervals. As such, locations and time have hierarchical structures inherently.

Furthermore, from observations of SBD applications as described later in this paper, they are hybrid processes consisting of data management and data management. In SBD applications, more time is often observed to be spent on development and execution of data management including preprocessing and postprocessing in addition to data mining than on data management.

B. Reproducibility

In general, the validity of published results of scientific researches has recently been judged based on not only traditional peer reviews but also reproducibility [5]. Reproducibility means that the same results with reported ones can be obtained by independent researchers. Success of reproduction hinges on detailed descriptions of methods and procedures, as well as data which have led to the published results.

At an extreme end of reproducibility spectrum [6] [7] is repeatability. Repeatability in computer science means that independent researchers can obtain exactly the same results by using the same data and the same codes that the reporters used. However, it is not always possible to use the same data and programming codes due to several reasons, such as limited space for publishing research results and lack or delayed spread of related standardization. Rex [8] is among ambitious attempts to facilitate repeatability. Rather, reproduction is done in order to make certain the essence of the experiments. All this is true of SBD researches and developments.

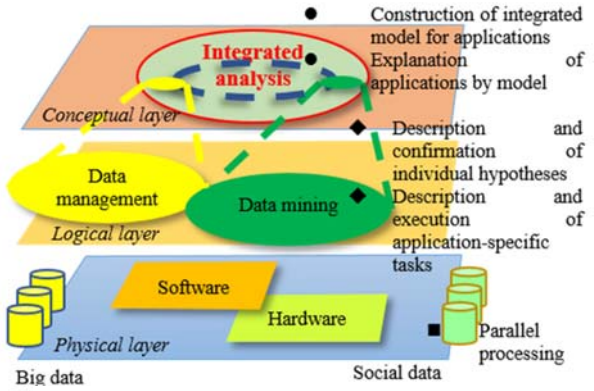

Figure 1. The reference architecture for social big data.

C. Data Model Approach

Reproducibility in researches and developments of SBD applications requires at least the following requirements.

- Description of SBD applications must be as independent from individual programming languages and frameworks as possible. It is not always possible for all researchers to access the same data and tools that the authors have used. However, reproducibility can be realized by enabling mappings from description of applications by an *SBD model* proposed in this paper to individual tools available for the other researchers even if the tools are not the same with the original one. Therefore, an SBD model (i.e., at conceptual level) must be more abstract than programming codes (i.e., at logical level). Such an abstract model is also expected to reduce the amount of description. In addition, the descriptions must be even as independent from programing models, such as parallel computing as possible. This *SBD model* approach can lead to increase of accountability of SDB applications to stake holders including generic users as a solution to the fourth V (i.e., vagueness).
- Both data management and data mining must be described in an integrated manner. Further, data management and data mining cannot be always separated in a crisp manner. Rather, in most SBD applications data mining is intermingled with data management. Later, such examples will be described in case studies. Therefore, an SBD model must be able to describe data management and data mining seamlessly.

We propose an integrated data model for describing SBD applications as hybrid processes of data management and data mining. And we describe applications using the model

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

as case studies for the validation of the proposed model. The reference architecture for SBD is illustrated in Figure 1.

D. Relation with Other Work

This paper extends our previous work [1] to enhance the validation of the proposed model with respect to a variety of case studies. To our knowledge, there are no abstract data models that can handle data management and data mining. Relational models [9] and object models [10] are dedicated data models solely for data management. Indeed, there exist a lot of programming languages and frameworks that can host data management and data mining, such as Spark [11] and MLI [12]. However, such language interfaces have different levels of abstraction from those of our data model, which is proposed in this paper. Rather, those are among candidate targets to which our abstract model can be translated.

III. DATA MODEL FOR SBD

A. Overview

We propose an abstract data model as an approach to reinforcing both reproducibility and accountability of SBD. Our SBD model aims to satisfy the following requirements:

- Enable to describe data management and data mining in an integrated fashion or seamlessly.
- Be independent enough from existing programming languages and frameworks and easy enough to translate into executable programming languages, as well.

First, we extend relational model [9] prevalent in data management fields as our approach to SBD model. The Relational model is based on a mathematical concept of a *set*. On the other hand, data mining includes clustering, classification, and association rules [4]. Clustering partitions a given set of data into a collection of sets, each of which has elements similar to each other. Classification divides a given set of data into pre-scribed categories or classes, that is, a collection of sets by using supervised learning. Association rule mining discovers a collection of frequent itemsets collocating in transactional data. Unlike the relational model, all these data mining techniques handle a *collection of sets* instead of a set. In other words, we must bridge gaps between data management and data mining with respect to levels of granularity.

Second, we would like to adopt abstractness comparable to those that relational models [9] and object models [10] have because such abstractness hides unnecessary details from the stakeholders and helps them to understand how the related data are used. We would also like to keep our model more abstract than individual programming languages and frameworks and to make the former translatable into the latter. In other words, our model is independent of programming models such as SQL (i.e., relational model) and NoSQL (e.g., Hadoop [13] and Crouchbase [14]). It is relatively easy to translate our model into SQL commands by facilitating conversions between families and sets (e.g., adding special columns representing indices to relations). Similarly, our model can be straightforwardly translated into NoSQL commands via SQL-like interfaces available for NoSQL, such as Hive [15] and N1QL [16].

We propose our SBD data model consisting of data structures and operations in the following subsections.

B. SBD Data Structures

Our SBD model uses a mathematical concept of a *family* [17], a collection of sets, as a basis for data structures. Family can be used as an apparatus for bridging the gaps between data management operations and data analysis operations.

Basically, our database is a *Family*. A Family is divided into *Indexed family* and *Non-Indexed family.* A Non-Indexed family is a collection of sets.

An Indexed family is defined as follows:

- $\{Set\}$ is a Non-Indexed family with *Set* as its element.
- $\{Set_i\}$ is an Indexed family with Set_i as its *i-th* element. Here *i: Index* is called *indexing set* and *i* is an element of Index.
- *Set* is {<*time space object*>}.
- Set_i is $\{<time\ space\ object>\}_i$. Here, *object* is an identifier to arbitrary identifiable user-provided data, e.g., record, object, and multimedia data appearing in social big data. *Time* and *space* are universal keys across multiple sources of social big data.
- $\{Indexed\ family_i\}$ is also an Indexed family with $Indexed\ family_i$ as its *i-th* element. In other words, Indexed family can constitute a hierarchy of sets.

Please note that the following concepts are interchangeably used in this paper.

- Singleton family ⇔ set
- Singleton set ⇔ element

In addition, Index of Indexed family can be used for implementing popular data structures such as array, vector, and matrix. In turn, matrix can represent graphs or networks frequently used for analyzing structures of social media such as Twitter [18].

As described later in Section IV, we can often observe that SBD applications contain families as well as sets and they involve both data mining and data management. Please note that a family is also suitable for representing hierarchical structures inherent in time and locations associated with social big data.

If operations constructing a family out of a collection of sets and those deconstructing a family into a collection of sets are provided in addition to both family-dedicated and set-dedicated operations, SBD applications will be described in an integrated fashion by our proposed model.

C. SBD Operations

SBD model constitutes an algebra with respect to Family as follows.

SBD is consisted of Family data management operations and Family data mining operations. Further, Family data management operations are divided into Intra Family operations and Inter Family operations.

1) Intra Family Data Management Operations

a) Intra Indexed Intersect (i:*Index Db* $p(i)$) returns a singleton family (i.e., set) intersecting sets which satisfy the predicate $p(i)$. Database *Db* is a Family,

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

which will not be mentioned hereafter.

b) Intra Indexed Union (*i:Index Db p*(*i*)) returns a singleton family union-ing sets which satisfy *p*(*i*).

c) Intra Indexed Difference (*i:Index Db p*(*i*)) returns a singleton family, that is, the first set satisfying *p*(*i*) minus all the rest of sets satisfying *p*(*i*)

d) Indexed Select (*i:Index Db p1*(*i*) *p2*(*i*)) returns an Indexed family with respect to *i* (preserved) where the element sets satisfy the predicate *p1*(*i*) and the elements of the sets satisfy the predicate *p2*(*i*). As a special case of true as *p1*(*i*), this operation returns the whole indexed family. In a special case of a singleton family, Indexed Select is reduced to Select (a relational operation).

e) Indexed Project (*i:Index Db p*(*i*) *a*(*i*)) returns an Indexed family where the element sets satisfy *p*(*i*) and the elements of the sets are projected according to *a*(*i*), attribute specification. This also extends also relational Project.

f) Intra Indexed cross product (*i:Index Db p*(*i*)) returns a singleton family obtained by product-ing sets which satisfy *p*(*i*). This is extension of Cartesian product, one of relational operators.

g) Intra Indexed Join (*i:Index Db p1*(*i*) *p2*(*i*)) returns a singleton family obtained by joining sets which satisfy *p1*(*i*) based on the join predicate *p2*(*i*). This is extension of join, one of relational operators.

h) Sort (*i:Index Db p*(*i*) *o*()) returns indexed family where the element sets satisfy *p*(*i*) and the elements of the sets are ordered according to the compare function *o*() with respect to two elements.

i) Indexed Sort (*i:Index Db p*(*i*) *o*()) returns an indexed family where the element sets satisfy *p*(*i*) and the sets are ordered according to *o*(), the compare function with respect to two sets.

j) Select-Index (*i:Index Db p*(*i*)) returns *i:Index* of set_i which satisfy *p*(*i*). As a special case of true as *p*(*i*), it returns all index.

k) Make-indexed family (*Index Non-Indexed Family*) returns an indexed Family. This operator requires *order-compatibility*, that is, that *i* corresponds to *i-th set of Non-Indexed Family*.

l) Partition (*i:Index Db p*(*i*)) returns an Indexed family. Partition makes an Indexed family out of a given set (i.e. singleton family either w/ or w/o index) by grouping elements with respect to *p* (*i:Index*). This is extension of "groupby" as a relational operator.

m) ApplyFunction (*i:Index Db f*(*i*)) applies *f*(*i*) to *i-th* set of DB, where *f*(*i*) takes a set as a whole and gives another set including a singleton set (i.e., Aggregate function). This returns an indexed family. *f*(*i*) can be defined by users.

2) Inter Family Data Management Operations Index-Compatible

a) Indexed Intersect (*i:Index Db1 Db2 p*(*i*)) union-compatible

b) Indexed Union (*i:Index Db1 Db2 p*(*i*)) union-compatible

c) Indexed Difference (*i:Index Db1 Db2 p*(*i*)) union-compatible

d) Indexed Join (*i:Index Db1 Db2 p1*(*i*) *p2*(*i*))

e) Indexed cross product (*i:Index Db1 Db2 p*(*i*))

Indexed (*) operation is extension of its corresponding relational operation. It preserves an Indexed Family. For example, Indexed Intersect returns an Indexed Family whose element is intersection of corresponding sets of two indexed families *Db1* and *Db*2, which satisfy *p*(*i*). At this time, we impose *union-compatibility*. Further, in case both *Db1* and *Db*2 are singleton families and *p*(*i*) is constantly true, Indexed Intersect is reduced to Intersect, which returns intersection of two sets (a relational operation). Indexed Union and Indexed Difference are also similarly interpreted.

3) Family Data Mining Operations

a) Cluster (*Family method similarity* {*par*}) returns a Family as default, where Index is automatically produced. This is an unsupervised learner. In *hierarchical agglomerative clustering* or HAC, as well as similar methods, such as some spatial, temporal, and spatio-temporal clustering, index is merged into new index as clustering progresses. *method* includes k-means, HAC, spatial, temporal, etc. *similarity/distance* includes Euclidean, Cosine measure, etc. *par* (ammeters) depend on *method.*

b) Make-classifier (*i:Index set:Family learnMethod* {*par*}) returns a classifier (Classify) with its accuracy. This is a supervised learner. In this case *index* denotes classes (i.e., predefined categories). Sample *set* includes both training set and test set. *learnMethod* specifies methods, such as decision tree, SVM, deep learning. *par* (ammeters) depend on *learnMethod.* This operation itself is out of range of our algebra. In other words, it is a *meta*-operation.

c) Classify (*Index/class set*) returns an indexed family with class as its index.

d) Make-frequent itemset (*Db supportMin*) returns an Indexed Family as frequent itemsets, which satisfy *supportMin*.

e) Make-association-rule (*Db confidenceMin*) creates association rules based on frequent itemsets *Db*, which satisfy *confidenceMin*. This is out of range of our algebra, too.

The predicates and functions used in the above operations can be defined by the users in addition to the system-defined ones such as Count.

IV. CASE STUDY

A. Case One

We have validated our model by applying it to social big data applications consisting of data management and datamining, such as tourism and disaster prevention.

We use the following background colors and bounding box for each category of SBD operations for illustration:

- Relational (set) data management operation
- Family data management operation
- Data mining operation

First, we describe a case study, analysis of behaviors of foreigners (visitors or residents) in Japan [19]. This case is

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

classified as analysis based on a single source of social data.

To classify foreign users as *residents* or *visitors*, we will classify the length of the stay in the country of interest as *long* and *short*, respectively. We assume the target country (i.e., the country of interest) uses one language dominantly. We first obtain the tweets that a user posted in Japan. We detect the principal language of the user in order to extract only foreign Twitter users. We define the *principal language* of a user as the language that meets the following two conditions.

- The language must be used in more than half of all the user's tweets. Since the Language-Detection toolkit [20] is over 99% precision according to their claim in detecting the tweet language, we used this toolkit in the experiment.
- The language must be selected by the user in his/her account settings. This means that the user claims that they use that language.

If the resultant principal language for a Twitter user is a language other than the one dominantly used in the target country, we regard the user as a foreign Twitter user and then classify the user as residents or visitors.

First, we sort a user's tweets posted in the target country in chronological order, where t_i denotes i-th tweet. Next, we set parameters *start date* and *stop date*, which specify the start and end date of interest, respectively. We define the oldest tweet between *start date* and *stop date* as T_{old} and define the parameter *travel period* as the maximum length of the stay. We define the newest tweet between T_{old} and T_{old} + *travel period* as T_{new}. Also, we set parameter j, a margin that ensures the foreign user is out of the target country. We identify a foreign user's tweets during a visit, if and only if all his/her tweets satisfy the following conditions:

- The foreign user posts more than T_{min} tweets between T_{old} and T_{new} and the user posts no tweets during the period from j days before to T_{old} to and the period from T_{new} to j days after T_{new}.

Here, T_{min} is the minimum number of tweets to prevent misclassification owing to a small number of tweets. The tweets posted between T_{old} and T_{new} are identified as the tweets during the visit. Since some users repeatedly visit the target country, we repeat the identification of tweets during a visit after T_{new}.

A foreign user is identified as a *visitor* to the target country, if and only if all his/her tweets between *start date* and *stop date* are tweets during visits. Foreign users who are not visitors are identified as *residents*. Here, we excluded foreign users who *tweeted* equal to or less than T_{min} times between *start date* and *stop date* as *unrecognizable*.

We obtained 72, 059, 720 geo-tagged tweets posted in Japan from Jun. 11, 2014 to Dec. 20, 2014 and used them as the dataset.

First, $\text{Classify}_{\text{foreign/domestic}}$ ({*Foreign Domestic*} DB_{tweet}) binarily splits into foreign and domestic sets as *AccountOrigin* (i.e., index class).

This classifier is based on a heuristic (i.e., manually-coded) rule for deciding foreigner as follows:

if Count (*t:tweet t.AccountId=AccountId* & *t.DetectedLanguage*()=*t.AccountLanguage* & *t.AccountLanguage*<>"Japanese") >=0.5*Count (t:tweet *t.AccountId=id*) then return foreign else domestic

The following fragment of descriptions collects only tweets posted by foreigners ("←" is the assignment operator):

DB_t ← Sort (Select (DB_{tweet} *Time of Interest* & Within "*Japan"*) *compare-time*()) ; singleton family (i.e., set).
$DB_{foreign}$ ← Indexed-Select ($\text{Classify}_{\text{foreign/domestic}}$ ({*Foreign Domestic*} DB_t) *AccountOrigin="foreign"*); singleton family.

Next, $\text{Classify}_{\text{visitor/resident}}$ ({*Visitor Resident*} DB_{tweet}) , a data mining operator, binarily splits into visitor and resident sets as *AccountStatus* (i.e., index class).

This is based on a heuristic rule for deciding inbound visitor as follows:

if Count (*t.tweet t.AccountId=AccountId* & T_{old}=<*t.time*=<T_{new})>=C_{min} & *Count* (*t.tweet t.AccountId=id* & T_{old}-j<=*t.time*<T_{old})=0 & Count (*t.tweet t.AccountId=id* & T_{new}<*t.time*<=T_{new}+j)=0 then return visitor else resident

The following fragment classifies tweets by foreigners into ones by inbound visitors and ones by foreign residents:

$DB_{foreignVisitorOrResident}$ ← $\text{Classify}_{\text{visitor/resident}}$ ({*Visitor Resident*} $DB_{foreign}$); This returns an indexed family.
$DB_{visitor}$ ← Indexed-Select ($DB_{foreignerVisitorOrResident}$ *AccountStatus*="*visitor*") ; This returns a singleton family.
$DB_{resident}$ ← Indexed-Select ($DB_{foreignerVisitorOrResident}$ *AccountStatus*="*resident*"); This returns a singleton family.

B. Case Two

Next, we describe another case study, finding candidate access spots for accessible Free Wi-Fi in Japan [21]. This case is classified as integrated analysis based on two kinds of social data.

This section describes our proposed method of detecting attractive tourist areas where users cannot connect to accessible Free Wi-Fi by using posts by foreign travelers on social media.

Our method uses differences in the characteristics of two types of social media:

Real-time: Immediate posts, e.g., Twitter

Batch-time: Data stored to devices for later posts, e.g., Flickr

Twitter users can only post tweets when they can connect devices to Wi-Fi or wired networks. Therefore, travelers can post tweets in areas with Free Wi-Fi for inbound tourism or when they have mobile communications. In other words, we can obtain only tweets with geo-tags posted by foreign travelers from such places. Therefore, areas where we can obtain huge numbers of tweets posted by foreign travelers are identified as places where they can connect to accessible Free Wi-Fi and /or that are attractive for them to sightsee.

Flickr users, on the other hand, take many photographs by using digital devices regardless of networks, but whether they

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

can upload photographs on-site depends on the conditions of the network. As a result, almost all users can upload photographs after returning to their hotels or home countries. However, geo-tags annotated to photographs can indicate when they were taken. Therefore, although it is difficult to obtain detailed information (activities, destinations, or routes) on foreign travelers from Twitter, Flickr can be used to observe such information. In this study, we are based on our hypothesis of "A place that has a lot of Flickr posts, but few Twitter posts must have a critical lack of accessible Free Wi-Fi." We extracted areas that were tourist attractions for foreign travelers, but from which they could not connect to accessible Free Wi-Fi by using these characteristics of social media. What our method aims to find is places currently without accessible Free Wi-Fi.

There are two main reasons for areas from where foreign travelers cannot connect to Free Wi-Fi. The first is areas where there are no Wi-Fi spots. The second is areas where users can use Wi-Fi but it is not accessible. We treat them both the same as inaccessible Free Wi-Fi because both areas are unavailable to foreign travelers. Since we conducted experiments focused on foreign travelers, we could detect actual areas without accessible Free Wi-Fi. In addition, our method extracted areas with accessible Free Wi-Fi, and then other locations were regarded as regions currently without accessible Free Wi-Fi.

This subsection describes a method of extracting foreign travelers using Twitter and Flickr. We obtained and analyzed tweets posted in Japan from Twitter using Twitter's Streaming application programming interface (API) [18]. We used the method introduced in Case One to extract foreign travelers.

Figure 2. High density areas of tweets (left) and of Flickr photos (right).

We obtained photographs with geo-tags taken in Japan from Flickr using Flickr's API [22]. We extracted foreign travelers who had taken photographs in Japan. We regard Flickr users who had set their profiles of habitation on Flickr as Japan or associated geographical regions as the users *living* in Japan; otherwise, they are regarded as foreign *visitors*. We used the tweets and photographs that foreign visitors had created in Japan in the analysis that followed. Our method envisaged places that met the following two conditions as candidate access spots for accessible free Wi-Fi:

- Spots where there was no accessible Free Wi-Fi
- Spots that many foreign visitors visited

We use the number of photographs taken at locations to extract tourist spots. Many people might take photographs of subjects, such as landscapes based on their own interests. They might then upload those photographs to Flickr. As these were locations at which many photographs had been taken, these places might also be interesting places for many other people to sightsee or visit. We have defined such places as tourist spots. We specifically examined the number of photographic locations to identify tourist spots to find locations where photographs had been taken by a lot of people. We mapped photographs that had a photographic location onto a two-dimensional grid based on the location at which a photograph had been taken to achieve this. Here, we created individual cells in a grid that was 30 square meters. Consequently, all cells in the grid that was obtained included photographs taken in a range. We then counted the number of users in each cell. We regarded cells with greater numbers of users than the threshold as tourist spots.

The fragment collects attractive tourist spots for foreign visitors but without accessible free Wi-Fi currently (See Figure 2):

$DB_{t/visitor}$ ← Tweet DB of foreign visitors obtained by similar procedures like case one (i.e., hybrid processes consisting of data management and data mining);

$DB_{f/visitor}$ ← Flickr photo DB of foreign visitors obtained by similar procedures like case one;

T ← Partition (*i:Index grid* $DB_{t/visitor}$ *p*(*i*)); This partitions foreign visitors tweets into grids based on geo-tags; This operation returns a indexed family.

F ← Partition (*j:Index grid* $DB_{f/visitor}$ *p*(*j*)); This partitions foreign visitors photos into grids based on geo-tags; This operation returns a indexed family.

Index1 ← Select-Index (*i:Index T Density*(*i*) >= *th1*); *th1* is a threshold. This operation returns a singleton family.

Index2 ← Select-Index (*j:Index F Density*(*i*) >= *th2*); *th2* is a threshold. This operation returns a singleton family.

Index3 ← Difference (*Index2 Index1*); This operation returns a singleton family.

We collected more than 4.7 million data items with geo-tags from July 1, 2014 to February 28, 2015 in Japan. We detected tweets tweeted by foreign visitors by using the method proposed by Saeki *et al.* [19]. The number of tweets that was tweeted by foreign visitors was more than 1.9 million. The number of tweets that was tweeted by foreign visitors in the Yokohama area was more than 7,500. We collected more than 5,600 photos with geo-tags from July 1, 2014 to February 28, 2015 in Japan. We detected photos that had been posted by foreign visitors to Yokohama by using our proposed method. Foreign visitors posted 2,132 photos. For example, grids indexed by *Index3* contain "Osanbashi Pier." Please note that the above description doesn't take unique users into consideration.

C. Case Three

We use another tourism-related work [23] for a third case, which is classified as integrated analysis based on both social data and open data.

Tourists want real-time information and local unique seasonal information posted on web sites, according to a survey study of IT tourism and services to attract customers by the Ministry of Economy, Trade and Industry (METI) [24].

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Current web sites provide related information in the form of guide books. Nevertheless, the information update frequency is usually low. Because local governments, tourism associations, and travel companies provide information about travel destination independently, it is difficult for tourists to collect "now" information for tourist spots.

Therefore, providing current, useful, real-world information for travelers by capturing the change of information in accordance with the season and time zone of the tourism region is important for the travel industry. We define "now" information as information for tourism and disaster prevention necessary for travelers during travel, such as best flower-viewing times, festivals, and local heavy rains.

We propose a method to estimate the best time for phenological observations for tourism such as the best-time viewing cherry blossoms ("Sakura" in Japanese) and autumn leaves in each region by particularly addressing phenological observations assumed for "now" information in the real world. Tourist information for the best time requires a peak period to view blooming flowers. Furthermore, the best times differ depending on regions and locations. Therefore, it is necessary to estimate a best time of phenological observation for each region and location. Estimating the best-time viewing requires the collection of a lot of information having real-time properties. For this research, we use Twitter data obtained for many users throughout Japan.

Preprocessing includes reverse geocoding and morphological analysis, as well as database storage for collected data.

Reverse geocoding identifies prefectures and municipalities by town name from latitude and longitude information of the collected tweets. We use a simple reverse geocoding service [25] available from the National Agriculture and Food Research Organization in this process: e.g., (latitude, longitude) = (35.7384446, 139.460910) by reverse geocoding becomes (Tokyo, Kodaira, City, Ogawanishi-cho 2-chome).

The standard lengths of time we used for the simple moving average were a seven-day moving average and one-year moving average. Since geo-tagged tweets tend to be more frequent at weekends than on weekdays, a moving average of seven days (one-week periodicity) is taken as one of estimation criteria. And the phenomenological observation is based on the one-year moving average as the estimation criterion since there are many "viewing" events every year, such as "viewing of cherry blossoms", "viewing of autumn leaves" and "harvesting period."

Table I: Examples of the target word.

Items	Target Words	In English
さくら	桜, さくら, サクラ	Cherry blossoms
かえで	楓, かえで, カエデ	Maple
いちょう	銀杏, いちょう, イチョウ	Ginkgo
こうよう	紅葉, 黄葉, こうよう, もみじ, コウヨウ, モミジ	Autumn leaves

Morphological analysis divides the collected geo-tagged tweet into morphemes. We use the *Mecab* morphological analyzer [26]. For example, the text "桜は美しいです" ("Cherry blossoms are beautiful" in English) is divided into morphemes ("桜 cherry blossom" / noun), ("は -" / particle), ("美しい beautiful" / adjective), ("です is" / auxiliary verb), and ("。." / symbol).

We use the simple moving average calculated by the formula (1) using the number of data going back to the past from the day before the estimated date of the best-time viewing.

$$X(Y) = \frac{P_1 + P_2 + \cdots + P_Y}{Y} \quad (1)$$

$X(Y)$: Y day moving average
P_n: Number of data of n days ago
Y: Calculation target period

Our method for estimating the best-time viewing processes the target number of extracted data and calculates a simple moving average, yielding an inference of the best flower-viewing time. The method defines a word related to the best-time viewing, estimated as the target word. The target word can include Chinese characters, hiragana, and katakana, which represents an organism name and seasonal change, as shown in Table I.

Next, we calculate a moving average for the best-time viewing judgment. The method calculates a simple moving average using data aggregated on a daily basis by the target number of extraction data. Figure 3 presents an overview of the simple moving average of the number of days.

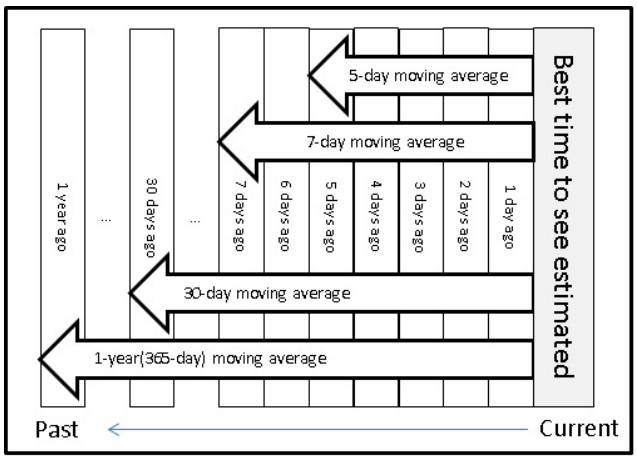

Figure 3. Number of days simple moving average.

In addition to the seven-day moving average and the one-year moving average, we also use the moving average of the number of days depending on each phenological target. In this study, we set the number of days of moving average from specified biological period of phenological target.

As an example, we describe cherry blossoms. The Japan Meteorological Agency (JMA) carries out phenological observations of Sakura, which yields two output the

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

flowering date and the full bloom date of observation target [27]. JMA uses one specimen tree for observations in each target area, which produces open data. The flowering date of Sakura is the first day of blooming 5–6 or more wheels of flowers of a specimen tree. The full bloom date of Sakura is the first day of a state in which about 80% or more of the buds are open in the specimen tree. In addition, for Sakura the number of days from flowering until full bloom is about 5 days. Therefore, Sakura in this study uses a five-day moving average as a standard.

If the number of tweets on each day exceeds the one-year moving average, the Condition 1 holds as follows.

$$P_1 \geqq X(365) \quad (2)$$

For the Condition 2, we use the following inequation with respect to both biological moving average dependent on species and seven-day moving average for one-week periodicity.

$$X(A) \geqq X(B) \quad (3)$$

In case of cherry blossoms, we set A and B to five days and seven days, respectively. This inequation determines the date on which the moving average of a short number of days exceeds the moving average of a long number of days.

The Condition 2 holds if the inequality 3 continuously holds for the number of days, which is made equal to or more than half of the moving average of a short number of days. In the case of cherry blossoms, 5 days / 2 = 2.5 days≒ 3 days is the consecutive number of days. That is, if the 5-day moving average exceeds the seven-day moving average by 3 days or more, it shall be the date satisfying the Condition 2.

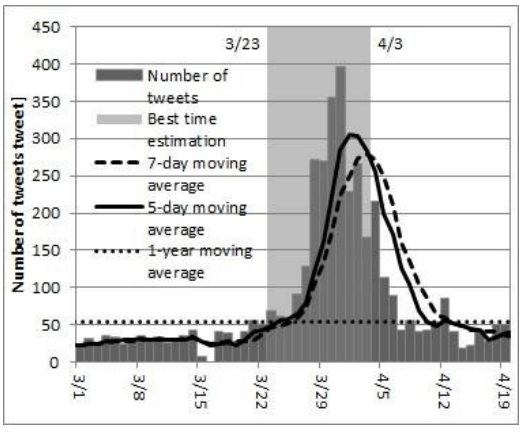

Figure 4. Results of the best time to see, as estimated for Tokyo in 2015.

We collected geo-tagged tweets posted in Japan from 2015/2/17 to 2016/6/30. The data include about 30 million items. We conducted the estimation experiment to ascertain the best-time viewing cherry blossoms uses the target word in Table I: "Sakura" or “cherry blossom,” which has a variety of spellings such as “桜” , “さくら” and “サクラ” in Japanese. The experimental target area is Tokyo. Our proposed method determines the best-time viewing duration in 2015.

The Condition 1 holds on the day when a dark gray bar exceeds the dotted line, one-year average (See Figure 4). The Condition 2 holds on the day when the solid line five-day average exceeds the broken line seven-day average for more than 3 days. Therefore, we estimated the duration 3/23 – 4/3 each of which satisfies both the Condition 1 and the Condition 2 as best-time viewing indicated by light gray area according to the following method:

ApplyFunction (ApplyFunction (Partition
(Select DB_{tweet} *Date* (2014, 2015) Within *“Tokyo”* *MorphologicalAnalyzer* (*Text*) contains (*Sakura*))
Date) Count) *Best-time-viewing-time* sequential

Here the function *Best-time-viewing-time* determines the duration as a temporal data mining operator, which is applied to one element of the set of counts of tweets per day after another as specified by the option *sequential*.

D. Case Four

We explain another case for integrated analysis based on both social data and open data.

For example, what in Osaka is equivalent to Tokyo Tower in Tokyo? Or what in Kobe corresponds to the Bay Bridge in Yokohama? Quantitatively answering these questions would be useful for applications such as destination management, which promotes travelers' visits to multiple touring spots in a region. In order to extract features of touristic resource names such as Tokyo Tower, we use Word2Vec [28] as a machine learning operator, which is suitable for natural language processing.

So we made an integrated hypothesis as follows.

Integrated hypothesis: By learning the vocabulary from corpus consisting of texts appearing in tweets (social data) containing a lot of touristic resource names such open data provided by the Ministry of Land, Infrastructure and Transport (MLIT) of Japan in consideration of the proximity of each other on the text, features of such names can be extracted. Furthermore, arithmetic operations such as plus and minus can be done over touristic resource names.

To this end, we collected geo-tagged tweets posted in Japan from March 11th to October 28th, 2015. Please note that we do not use location information in this case.

First, we create a list of touristic resource names by referring touristic resource data including open data of MLIT, and data obtained from TripAdvisor [29]. From the collected tweets, we selected about 115,610,000 tweets with touristic resource names on the list and the same number of tweets without them. Further we chose 500 tweets with touristic resource names and the same number of tweets without them to learn word semantics by Word2Vec as follows:

ApplyFunction (Union
(Select DB_{tweet} *MorphologicalAnalyzer*(*text*) contains (*touristic_resource_name*) fetch (500))
(Select DB_{tweet} *MorphologicalAnalyzer*(*text*) not contains (*touristic_resource_name*) fetch (500)))
Word2Vec

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Here the user-defined function MorphologicalAnalyzer creates a bag of words from each text. Then ApplyFunction produces an indexed family. Word2Vec is based on a simple neural network (See Figure 5). It consists of three layers: input, hidden, and output. Each weight vector in the hidden layer represents a representation of its corresponding word. The analyst specifies the dimension of the middle layer. The dimension corresponds to the number of neurons. In learning, in the output corresponding to the input of a word, the weight of the middle layer is calculated so that the probability of a word appearing near the input word becomes high. We consider this weight as a feature vector of each word. Linear operations such plus, minus can be made between feature vectors. In the experiment, we set parameters for Word2Vec such as a dimension of 400, Skip-gram, Hierarchical Soft Max, and a window size of 5. For the similarity of word vectors, cosine similarity is used in our experiments.

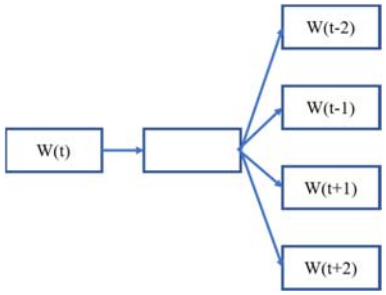

Figure 5. A neural network as Skip-gram model.

For example, we could calculate "Nankincho" by setting X = "Yokohama Chinatown" in the expression X - "Yokohama" + "Kobe."

Similarly, we could find “Akashi Kaikyo Bridge” by setting X = "Yokohama Bay Bridge" in the same expression. We have successfully confirmed the validity of the results by using relevant web pages.

E. Case Five

The last case is classified as integrated analysis based on a single source of social data and several sources of open data.

The 2011 Tohoku earthquake and tsunami showed the importance of smooth evacuation to nearest safe places. Smooth evacuation from crowded areas like train stations is difficult because people facing disaster often cannot find good paths to the nearest safe places and their evacuation may create congestion, which again hinders smooth evacuation. For smooth evacuation, the safety of roads used for evacuation needs to be evaluated. This evaluation requires 1) geographical characteristics of roads such as the collapse risk of neighboring buildings and road width and 2) crowdedness of roads at the moment of disaster. While previous studies considered the former, the latter information has not been well studied because the crowdedness depends on the demographics of a city, which is quite dynamic and difficult to measure. For example, daytime and nighttime populations of a city differ greatly. Our relevant work [30] proposes a method that measures demographics snapshot of a city from time and geo-stamped micro-blog posts and visualizes high-risk evacuation roads based on geographical characteristics and the demographics. Our method enabled visualization of high-risk evacuation roads on a per-hour basis. We also qualitatively analyze and discuss the visualization results.

Even though evacuees' first safe destinations are determined, we need to enable them to safely reach the destinations. To this end, local governments also specify evacuation roads, that is, roads considered to be safe and preferable for evacuation and recommended to be used. These roads are determined by considering three factors: 1) toughness of adjacent buildings to consider the risk of their collapse, 2) road width to consider the amount of people the roads can transport, and 3) population of adjacent areas to consider how crowded the roads become in the case of earthquakes. Evacuation roads and sites are manually revised about every five years.

Among the three factors, however, population of areas adjacent to roads is difficult to investigate because the actual population of a large city bearing millions of commuters is dynamic. For example, while the city center is densely populated during daytime, it is sparsely populated during nighttime. We thus need to take snapshots of dynamic demographics of a city. To tackle this problem, local governments currently use the Person Trip investigation [31] for determining evacuation roads. This is a government-led investigation by means of sending and gathering questionnaires to and from randomly sampled postal addresses. The questionnaire asks respondents where and how they usually go. However, these investigations are too costly to perform frequently, and the last one was performed in 1999 in Tokyo. In addition, they investigate only small samples of the whole population. Thus, infrequent and small samples in population snapshots make evaluation of crowdedness unreliable.

To determine roads preferable for evacuation, far more fine-grained demographics of areas of a city is needed considering the relative positions of each area and evacuation sites. Therefore, we propose a method for discovering high-risk paths (the roads for which evacuation risk is high), considering the number of people and geographical characteristics. To extract the number of people in each region, we use Twitter because it has many users and is a typical micro-blog, and we can understand “when” and “where” users upload tweets because they are tagged with time and location.

For example, people evacuating from crowded areas to evacuation sites are more likely to suffer fatal accidents such as stampedes and become confused than those evacuating from uncrowded areas. This means that evacuation from crowded areas is difficult. In addition, evacuation from crowded areas makes other roads crowded. Orange broken arrows in Figure 6 show crowded roads. Here we consider two types of crowded roads as depicted in Figure 6: Pattern A roads are made when people gather from two or more crowded areas, and Pattern B roads are made when a road is on the paths from a crowded area to multiple evacuation sites. We assume that crowded areas and crowded roads make evacuation difficult. In addition, geographical characteristics, such as many wooden buildings and many narrow roads

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

located in the areas or along the roads, also make evacuation difficult.

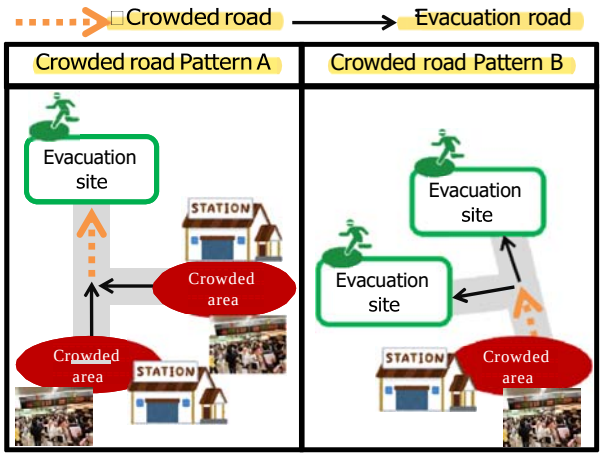

Figure 6. Examples of crowded roads.

1) Extraction of Crowded Areas and Evacuation Sites

When disasters occur, the situations in crowded areas are expected to be greatly influenced by people's behavior, which depends on the time of day. Therefore, we extract crowded areas using tweets with geo-tag every hour. First, we map tweets with geo-tag into two dimensional grids whose resolution is around 500 meters. Then, we adjust parameters for the extraction. Consequently, each cell in the grids includes tweets taken in the range. Then, using the grids, we calculate the number of tweets in each cell and normalize those numbers within a cell using the Gaussian Filter. The motivation of using a Gaussian filter here is to capture the fact that Twitter users move around after posting tweets and to allow the positioning errors of latitudes and longitudes obtained from GPS.

After the filtering, we extract cells in which tweets including photographs exceed the pre-determined threshold.

We make an index consisting of a crowded area c and an evacuation area e by using the dedicated function as follows:

$INDEX_{(c,\ e)} \leftarrow$ *Make-crowded-evacuation-area* (DB_{tweet}, *Gaussian*)

2) Extraction of Multiple Paths

We search multiple paths between evacuation sites and crowded areas extracted in the previous step. A path is defined as a route between an evacuation site and a crowded area. Road data consisting of routes are obtained from Open Street Map (OSM) [32]. In this work, we use pgRouting [33] to extract multiple paths for a crowded area c and a specified evacuation site e.

We extract paths from c to e as follows:

$FAMILY_{path} \leftarrow$ Make-indexed family ($INDEX_{(c,\ e)}$ *pgRouting*)

3) Extraction of High-Risk Path

We describe the method to extract high-risk paths from the paths obtained in the previous step. Our approach extracts high risk paths based on congestion degree and Emergency Response Difficulty Assessment (ERDA) degree. Congestion degree is considered as the weight of a crowded road and area as described below. ERDA degree is calculated based on geographical characteristics such as road width, possible degree of collapse of buildings, and fire risk. Finally, we calculate Risk degree using these measures. We define path l whose Risk degree is high as a High-Risk path.

a) Calculation of congestion degree

We calculate Congestion degree using crowded roads and the weight of a crowded area. We calculate betweenness centrality at target evacuation in 3km. We define the node located nearest to the median point of crowded areas on the network as node i (Origin) and define the node located nearest to each evacuation site as node j (Destination). Let g_{ij} be the number of paths between node i and node j. Also, we define $g_{ij}(l)$ as the number of occurrences of path l between node i and node j. Here, betweenness centrality $B(l)$ of path l can be written in the following Formula (4).

$$B(l) = \sum_{i \neq j} \sum_{j \in V_i, j \neq i} \frac{g_{ij}(l)}{g_{ij}} \quad (4)$$

V_i is the set of nodes of the evacuation sites that can be reached in less than 3km from node i.

Next, we describe a method for incorporating the weight of the crowded area into the calculation. Here, a weight is the estimated number of people passing through the path l. We incorporated the number of people extracted in the crowded area C, which is defined as R_C; the total number of evacuation sites from crowded area C within 3km, which is defined as $S(C)$; and the number of paths from crowded area C to evacuation site S, which is defined as $k(CS)$. After searching paths connected to crowded area C and evacuation site S, if path l is contained in the resulting paths, we calculate the weight of a crowded area, $W(l)$, using Formula (5).

$$W(l) = \sum_{C=1,N} \sum_{S=1,V(s)} \frac{R_C}{S(C) \cdot k(CS)} \quad (5)$$

Here, $V(S)$ is the set of nodes of evacuation sites that can be reached from the crowded area C within 3km. N is the total number of crowded areas extracted in the specified hour. Here, Congestion degree of path l can be written as Formula (6) by normalizing the results of Formula (4) and Formula (5).

$$Congestion(l) = Normalized(B(l)) + Normalized(W(l)) \quad (6)$$

b) Calculation of emergency response difficulty assessment degree

We describe a method for calculating Emergency Response Difficulty Assessment (ERDA) degree, another measure of risk of high-risk paths. ERDA degree is determined by the Bureau of Urban Development, Tokyo Metropolitan Government. The degree can be used as a measure of the difficulty of people's activities in disaster situations, and it comprehensively considers the building collapse probability, fire probability, and road width. The measure gives each district of each town of Tokyo in five-scale ranks. The lower ranks mean that people can more

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

smoothly evacuate from the district during disasters.

When calculating the ERDA degree, we first find the rank values of the districts with which path l overlaps. Then, we calculate the length of the path l within each district. Finally, we calculate the resulting assessment by multiplying the rank and the length of the path. This can be formally written as follows. Let path l have total length L and l overlap with the district whose stage is *Rank*. Let the total length of path l within the districts whose stage is *Rank* be L_{Rank}. Then, we define ERDA(l) as follows.

$$ERDA(l) = Normalized\left(\sum_{Rank=1}^{5} Rank \cdot L_{Rank}\right) \quad (7)$$

c) Calculation of risk degree

Finally, we describe a method for calculating Risk degree. Risk degree of path l, namely *Risk*(l), is defined as follows by using Formula (6) and Formula (7).

$$Risk(l) = Congestion(l) + ERDA(l) \quad (8)$$

The higher *Risk*(l) means that path l is more dangerous at the time of disasters.

We obtain Risk of paths $FAMILY_{path}$ by calculating the formulas (4)-(8) sequentiallly.

4) Experiments

We describe the data set used in this study. We collected 5,769,800 geo-tagged tweets posted in Tokyo's 23 wards, the center of Tokyo. We collected tweets from April 1, 2015 to December 31, 2015. The number of Twitter users that were identified was 235,942. Rather than directly using the number of people tweeting, we performed smoothing of the people's distribution using a Gaussian Filter and manually identified 250 unique twitter users per grid as a crowded area because the number of people per grid abruptly change on that boundary.

We also used road network data of Tokyo collected from OpenStreetMap [32]. Roads in this network data are tagged with types of roads. We excluded data tagged as *motorway*, *motorway_link*, and *motorway_junction* because we are focusing on people evacuating on foot, and people cannot walk those roads safely during a disaster. As a result, 205,930 nodes and 302,141 edges remain.

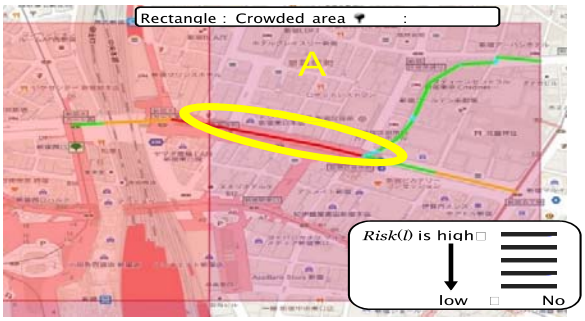

Figure 7. Visualization of high-risk paths Results from 5:00 am to 6:00 am.

a) Results from 5:00am to 6:00am

We describe high-risk paths extracted from 5:00am to 6:00am. All top-five high-risk paths extracted were parts of Yasukuni street within Shinjuku-Ward as shown by the red line within the yellow oval (A) in Figure 7.

From Figure 7, we can see that few lines are drawn around the oval (A) other than the straight red line. This result matches geographical characteristics of this area. While this area has many evacuation sites such as Shinjuku High School, Shinjuku Junior High School, and Tenjin Elementary School, evacuees can reach these sites without passing through narrow roads because this area is a city center where old narrow roads became obsolete and were demolished.

b) Results from 6:00pm to 7:00pm

We describe high-risk path extracted from 6:00pm to 7:00pm. The red line surrounded by the yellow circle and oval (B) of Figure 8 indicates 1st, 3rd, 4th, and 5th rankings. While the extracted areas from 5:00am to 6:00am and those from 6:00pm to 7:00pm share the same pattern (the areas around Shinjuku stations are detected to be crowded), our system successfully captured the difference in the overall patterns and different paths were extracted as high-risk paths.

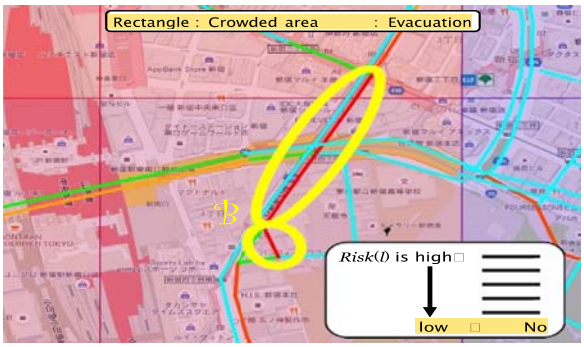

Figure 8. Visualization of high-risk paths Results from 6:00 pm to 7:00 pm.

V. CONCLUSION

We have proposed an abstract data model for integrating data management and data mining by using mathematical concepts of families, collections of sets. Our model facilitates reproducibility and accountability required for SBD researches and developments. We have partially validated our proposal by adapting our model to real case studies. Especially, we have illustrated that SBD applications are inherently mixed of data management and data mining and that integrated analysis based on different sources of data is effective in SBD applications. Technically, however, there still remain rooms to describe mappings from our model to existing programming tools, such as Spark. Further, we must devise some kinds of optimization comparable to query optimization of SQL. Empirically, we would like to validate our proposed model more thoroughly by adapting it to different kinds of applications.

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 16K00157, 16K16158, and Tokyo Metropolitan University Grant-in-Aid for Research on Priority Areas Research on social big data.

REFERENCES

[1] H. Ishikawa and R. Chbeir, "A data model for integrating data management and data mining in social big data," Proceedings of 9th IARIA International Conference on Advances in Multimedia (MMEDIA 2017), April 2017.

[2] IDC, *The Diverse and Exploding Digital Universe* (white paper, 2008). http:// www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf Accessed November 2017.

[3] IDC, *The Digital Universe In 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East* (2012). http://www.emc.com/leadership/digital-universe/iview/index.htm Accessed November 2017.

[4] H. Ishikawa, Social Big Data Mining, CRC Press, 2015.

[5] T. C. Südhof, "Truth in Science Publishing: A Personal Perspective," PLOS August 26, 2016.

[6] D. G. Feitelson, "From Repeatability to Reproducibility and Corroboration," ACM SIGOPS Operating Systems Review - Special Issue on Repeatability and Sharing of Experimental Artifacts, Volume 49, Issue 1, pp. 3-11, January 2015.

[7] R. D. Peng, "Reproducible Research in Computational Science," SCIENCE, VOL 334, 2, December 2011.

[8] S. Perianayagam, G. R. Andrews, and J. H. Hartman, "Rex: A toolset for reproducing software experiments," Proceedings of IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2010, pp. 613-617, 2010.

[9] R. Ramakrishnan and J. Gehrke, Database Management Systems, 3rd Edition, McGraw-Hill Professional, 2002.

[10] H. Ishikawa, Y. Yamane, Y. Izumida, and N. Kawato, "An Object-Oriented Database System Jasmine: Implementation, Application, and Extension," IEEE Trans. on Knowl. and Data Eng. 8, 2, pp. 285-304, April 1996.

[11] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica, "Apache Spark: a unified engine for big data processing," Com. ACM, 59, 11, pp. 56-65, October 2016.

[12] E. R. Sparks, A. Talwalkar, V. Smith, J. Kottalam, X. Pan, J. E. Gonzalez, M. J. Franklin, M. I. Jordan, and T. Kraska, "MLI: An API for distributed machine learning," Proceedings of the IEEE ICDM International Conference on Data Mining (Dallas, TX, Dec. 7–10). IEEE Press, 2013.

[13] Apache Hadoop. http://hadoop.apache.org/ Accessed November 2017.

[14] Crouchbase. https://www.couchbase.com/ Accessed November 2017.

[15] Apache Hive. http://hive.apache.org/index.html Accessed November 2017.

[16] N1QL. https://www.couchbase.com/products/n1ql Accessed August 2017.

[17] D. Smith, R. St. Andre, and M. Eggen, A Transition to Advanced Mathematics, Brooks/Cole Pub Co., 2014.

[18] Twitter, *Twitter Developer Documentation*. https://dev.twitter.com/streaming/overview Accessed November 2017.

[19] M. Hirota, K. Saeki, Y. Ehara, and H. Ishikawa, "Live or Stay?: Classifying Twitter Users into Residents and Visitors," Proceedings of International Conference on Knowledge Engineering and Semantic Web (KESW 2016), 2016.

[20] GitHub, Language Detection. https://github.com/shuyo/language-detection Accessed November 2017.

[21] K. Mitomi, M. Endo, M. Hirota, S. Yokoyama, Y. Shoji, and H. Ishikawa, "How to Find Accessible Free Wi-Fi at Tourist Spots in Japan," Volume 10046 of Lecture Notes in Computer Science, pp. 389-403, 2016.

[22] Flickr, *The App Garden*. https://www.flickr.com/services/api/ Accessed November 2017.

[23] M. Endo, S. Ohno, M. Hirota, Y. Shoji, and H. Ishikawa, "Examination of Best-time Estimation Using Interpolation for Geotagged Tweets," Proceedings of 9th IARIA International Conference on Advances in Multimedia (MMEDIA 2017), April 2017.

[24] Ministry of Economy, Trade and Industry, *study of landing type IT tourism and attract customers service*, http://www.meti.go.jp/report/downloadfiles/g70629a01j.pdf Accessed November 2017 (in Japanese).

[25] National Agriculture and Food Research Organization, *simple reverse geocoding service*, https://www.finds.jp/rgeocode/index.html.en Accessed November 2017.

[26] MeCab, *Yet Another Part-of-Speech and Morphological Analyzer*. http://taku910.github.io/mecab/ Accessed November 2017 (in Japanese).

[27] Japan Meteorological Agency, *Disaster prevention information XML format providing information page*. http://xml.kishou.go.jp/ Accessed November 2017 (in Japanese).

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*. 2013.

[29] TripAdvisor, *Read Reviews, Compare Prices & Book* http://tripadvisor.com Accessed November 2017.

[30] M. Kanno, Y. Ehara, M. Hirota, S. Yokoyama, and H. Ishikawa. "Visualizing High-Risk paths using Geo-tagged Social Data for Disaster Mitigation," 9th ACM SIGSPATIAL International Workshop on Location-Based Social Networks (LBSN '16), November 2016.

[31] Ministry of Land, Infrastructure, Transport and Tourism, *Results from the 4th Nationwide Person Trip Survey* http://www.mlit.go.jp/crd/tosiko/zpt/pdf/zenkokupt_gaiyoban_english.pdf Accessed November 2017.

[32] OpenStreetMap Contributors, *OpenStreetMap*. http://www.openstreetmap.org/ Accessed August 2017.

[33] pgRouting Community, *pgRouting*. http://pgrouting.org/ Accessed November 2017.

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org