Learning From Cross-Domain Media Streams for Event-of-Interest Discovery

Wen-Yu Lee, Winston H. Hsu, *Senior Member, IEEE*, and Shin'ichi Satoh, *Member, IEEE*

***Abstract*—Every day, vast amounts of data are uploaded to various social-sharing websites. Each social-sharing website has its own media dataset. Recently, mining media datasets has shown great potential for our daily lives, e.g., earthquake detection. Generally, different datasets have different characteristics. Combining different datasets is capable of achieving better performance than using any dataset independently, particularly if the datasets can compensate for each other. The resulting performance, however, depends on the fusion method. Effectively combining different datasets is challenging. As a solution to this challenge, this paper presents a generic two-stage framework for events of interest. Specifically, the first stage normalizes the contents of different datasets to make them comparable; then, the second stage combines the normalized contents for a ranked event list using graph-based algorithms. Practically, this paper unifies a flow-based media dataset and a check-in-based media dataset. Based on the precision for the top *n* events, the experimental results demonstrate that the proposed framework can achieve better performance in finding events associated with sports, local festivals, concerts, and exhibitions compared with a state-of-the-art approach that uses one dataset alone.**

***Index Terms*—Cross-domain media mining, event discovery, graph-based data fusion, social network analysis.**

I. Introduction

In RECENT years, social media have changed the world. Increasingly more people are sharing their daily lives through social media. The amount of media data uploaded to social-sharing websites has increased by millions per day. For example, more than 300 million images on average are uploaded to Facebook in a day [1]. With the increasing volume of media data, there have been many studies in the literature that aim to discover and/or summarize media datasets for useful information. The majority of previous works focused on mining a single media dataset for specific applications, e.g., a system for earthquake detection based on Twitter [2].

Generally, every media dataset has its own characteristics and information. For example, Kuo *et al.* observed that most Instagram users like to share information regarding food and travel, whereas many Twitter users prefer sharing information related to sports and news [3]. Furthermore, they conducted an experiment that compared the degree of overlap for popular locations from four media datasets, including Twitter, Instagram, TripAdvisor, and New York City open data, and their results did support their intuition. Consequently, they summarized the life aspects, e.g., trends, food, and fashion, of New York City based on the results from multiple media datasets. Similarly, Becker *et al.* also considered multiple media datasets, including Twitter, YouTube, and Flickr, and then provided diverse media contents for target events [4].

This paper unifies two media datasets for events of interest. Unifying media datasets is capable of reducing the effects of noisy data [5] and remedying the deficiencies of a single media dataset. In contrast to related works, e.g., [3], which explored multiple datasets separately for diverse aspects, this paper seamlessly combines a flow-based media dataset and a check-in-based media dataset for high search performance. The flow-based media dataset contains navigation information for some locations, e.g., the total number of people who enter or exit taxis per day at the Sapporo station. The check-in-based media dataset contains user-contributed information for some locations, e.g., comments and images. We used a flow dataset of taxis and a check-in dataset from Instagram. We collected data for the City of Sapporo from March to November in 2014, where the taxi dataset was created using GPS data from taxis. Note that we did not use another check-in-based media dataset, e.g., Twitter, because Instagram not only has textual data but also a large number of images for future studies. Furthermore, Instagram is more convenient for long-term data collection. However, the methods presented in this paper can easily be extended to many other check-in-based media datasets, e.g., Foursquare, which provides information such as textual data, images, and ratings that can also be used by the proposed methods.

Table I compares the two media datasets used throughout this paper.

As shown in this table, the characteristics of the two datasets are different and complementary. Both the taxi dataset and the Instagram dataset contain numerical data, i.e., numbers of entries and exits for the taxi dataset and the number of check-ins for

Manuscript received October 26, 2016; revised March 27, 2017 and June 7, 2017; accepted June 29, 2017. Date of publication July 12, 2017; date of current version December 14, 2017. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 103-2221-E-002-105-MY3 and Grant MOST 105-2917-I-564-054 and sponsored by Microsoft Research Asia. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Marco Bertini. *(Corresponding author: Wen-Yu Lee.)*

W.-Y. Lee is with the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 10617, Taiwan (e-mail: majorrei@cmlab.csie.ntu.edu.tw).

W. H. Hsu is with the Graduate Institute of Networking and Multimedia, Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: winston@csie.ntu.edu.tw).

S. Satoh is with the National Institute of Informatics, Tokyo 101-8430, Japan (e-mail: satoh@nii.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2017.2726184

TABLE I
COMPARISON OF THE FLOW DATASET OF TAXIS AND THE INSTAGRAM CHECK-IN DATASET FOR THE CITY OF SAPPORO

	Taxis (flow dataset)	Instagram (check-in dataset)
Information	time, places, values	time, places, texts images, values
Hotspots / Weak spots	weekdays / holidays	holidays / weekdays
Reaction	relatively instantaneous	relatively non-instantaneous
Distribution	relatively clustered	relatively scattered
Coverage of Locations	61% (by the check-in dataset)	47% (by the flow dataset)

the Instagram dataset for different locations for different dates. The numerical data for the taxi dataset are relatively more traceable because these data were collected from the same set of taxis. However, the taxi dataset does not contain textual data and images, which are quite beneficial for describing events. In addition, the hotspots of the two datasets are contrary to each other; thus, their data volumes for a day are complementary. Moreover, the reactions to events are instantaneous for the taxi dataset, but the reactions to events for the Instagram dataset are not sufficiently instantaneous (see Section V-D.1). Furthermore, the distributions of taxis' entry and exit locations are clustered, whereas the distributions of check-in locations are scattered. Approximately 61% of the entry and exit locations are close to some check-in locations, and approximately 47% of the check-in locations are close to some entry and exit locations (see Section V-C.3).

For event discovery, the taxi dataset offers real-time and a sufficient amount of numerical data, particularly on major roads. Observing changes in the data of this dataset provides a starting point for finding events of interest. However, understanding what the events are about is difficult due to the lack of textual information or visual information. In contrast, although the numerical data for the Instagram dataset are not as real time and sufficient as those for the taxi dataset, the Instagram dataset can not only provide textual information and even visual information but also extend the region of event discovery to a wider region. Because the two datasets are complementary, we intend to unify them for events of interest. However, combining datasets with different characteristics is challenging. Furthermore, the fusion method must be effective because the resulting performance directly depends on the fusion method used.

As a solution to the above challenge, we propose a generic two-stage framework for events of interest. Practically, in this paper, an event is defined as a gathering of people for a common purpose. Events of interest include sports, local festivals, concerts, and exhibitions. Although the proposed framework may be used for other events, such as graduations, these four types of events will be used for evaluating performance in our experiments. Given a flow dataset of taxis and the Instagram check-in dataset, the proposed framework will generate a ranked event list, where the locations of the top-ranked events are attractive in terms of large numbers of check-ins and large numbers of people who enter or exit taxis around the locations. The first stage of the framework separately normalizes given datasets, aiming to make data from different datasets comparable. Through graph construction, the second stage then fuses the normalized data from different datasets while considering the geographical information. Overall, the major contributions of this paper are summarized as follows:

1) To the best of our knowledge, this paper presents the first work to seamlessly combine a flow-based media dataset and a check-in-based media dataset for events of interest.
2) This paper presents a generic two-stage framework that normalizes data from different sources and then combines the normalized data using effective graph algorithms.
3) The experimental results show that our approach can on average achieve a 57% precision improvement by effectively combining two datasets compared with a state-of-the-art method that uses one dataset alone.

Note that in this paper, a location is a GPS point that has a fixed spatial coordinate, i.e., latitude and longitude. The remainder of this paper is organized as follows. Section II provides an overview of related works. Section III formulates the problem of finding events of interest. Section IV details the proposed approach. Section V evaluates the performance of the proposed approach. Finally, Section VI concludes this paper.

II. RELATED WORK

The majority of previous works focused on a single media dataset for event detection or event summarization. Based on Twitter tweets, Sankaranarayanan *et al.* constructed a news processing system to capture late breaking news [6]. Sakaki *et al.* constructed a reporting system for earthquake detection [2]. For trending topic detection, Aiello *et al.* observed that different approaches may greatly affect the quality of the results. Consequently, they implemented six topic detection approaches and then concluded that the use of n-gram co-occurrence and topic ranking can achieve the best performance [7]. For summarization, Weng *et al.* then constructed a system called Voters' Voice, which effectively summarized netizens' discussions for the Singapore General Election [8]. Recently, Meladianos *et al.* observed that some events typically evolve over time [9], e.g., natural disasters; thus, there could be a set of sub-events associated with the evolving events. Based on sub-event detection, they then proposed a real-time approach that can identify key moments and related tweets of events for event summarization. In summary, these works primarily focused on exploring information, particularly textual information, from a single media dataset. In contrast, this paper focuses on the method for fusing different media datasets.

Considering image datasets, Dao *et al.* analyzed the image collection of each specific event type, e.g., graduation, for an event signature and then presented a solution to associate image collections with different event types [10]. Believing that an image may belong to an event cluster, Ruocco and Ramampiaro presented a clustering approach that considers the tags, time, and geographic information of images. The clustering approach is expected to extract groups of images for related events from the given image dataset [11]. It is apparent that the tags of an image are critical for analyzing the event associated with the image. Some tags of an image might not be closely related to the image. Considering tag relevance, Shah *et al.* developed a tag ranking approach based on the voting results from photo neighbors [12]. In summary, these works considered visual information and/or textual information of images. In contrast, this paper does not use image information but rather uses textual information and taxis' traces.

Considering video datasets, Ma *et al.* developed an intermediate representation of videos, which is derived from the bag-of-words features of the videos, using target videos and external video archives [13]. Considering complex events in videos, Yan *et al.* discovered videos of particular events from Internet video archives [14]. Chang *et al.* studied video ranking for specified events where no training data are required [15]. Recently, Mazloom *et al.* represented videos as a set of tags, which are associated with events, and then propagated the tags to other videos for video event detection [16]. In summary, these works explored video datasets for specific applications. In contrast, this paper does not work on video datasets but rather presents a fusion method that consists of combining a flow-based media dataset and a check-in-based media dataset. Meanwhile, Zhang *et al.* identified the locations and times of events that occurred and the scales of the events based on a dataset of taxis' traces [17]. Compared with [17], this paper further considers a check-in-based media dataset, aiming to remedy the deficiencies of a dataset of taxis' traces. This paper determines not only the locations and times of events but also what the events are by leveraging the textual information from the check-in-based media dataset.

Considering multiple datasets, previous works have shown great potential for event summarization based on textual information, images, and/or videos from different datasets. Previously, Becker *et al.* discovered events from different datasets for different aspects of events [4]. Similarly, Kuo *et al.* investigated the characteristics of several datasets and then extracted information from the datasets to understand the daily lives of a city [3]. Later, Shah *et al.* concluded the challenges of event summarization on large-scale datasets, followed by developing a real-time summarization system leveraging the metadata of photos/videos from the given media dataset and the content of Wikipedia articles [18]. Shah *et al.* explored user-generated videos for mood recognition, followed by recommending users a ranked list of songs based on the predicted moods and their preferences [19]. In summary, these works showed the effectiveness of using multiple datasets for better summarization or recognition. In contrast, this paper discovers events of interest from media datasets and presents a framework that can be used to seamlessly combine different datasets.

Overall, this paper focuses on finding events of interest based on textual information and taxis' traces. Specifically, this paper develops a generic framework for fusing flow-based datasets and check-in-based datasets by normalizing data from different datasets, followed by combining data considering their geographic information.

III. Problem Formulation

This section presents the problem formulation. For clarity, we first define *a day* as the smallest unit of time for event discovery. However, the methods presented in this paper can be easily extended for a unit that is smaller than a day, e.g., an hour. In the following definition, we provide formal descriptions of two terms in this paper, i.e., *flow data* and *check-in data*.

Definition 1: **(Flow Data)**

A flow is a route from a particular source to a particular destination. Given a location for a period of time, flow data of the location can offer the statistics of some aspect per day at the location.

For example, for the flow dataset of taxis, flow data may offer the total number of people who enter or exit taxis per day at the Sapporo station.

Definition 2: **(Check-in Data)**

A check-in is a way for people to share their locations and descriptions about the locations. Given a location for a period of time, check-in data of the location can offer a set of tags and/or comments per day at the location from users of social networks.

Note that check-in data may offer more information than tags and comments, e.g., images. This paper will focus on textual information from tags and comments (although textual information could be extracted from images). Finally, our problem of finding events of interest in a region can be stated as follows.

Problem 1: Given a set of flow data of locations and a set of check-in data of locations for a period of time in a region, the objective of the problem of finding events of interest in a region is to create a ranked list of events in the region based on the given sets of data.

Practically, we used 1) a flow dataset of taxis and 2) an Instagram check-in dataset for events of interest in the City of Sapporo. In fact, these two datasets are not the original datasets but rather the resulting datasets after pre-processing for ease of use. Specifically, the original flow dataset of taxis consists of a list. Each item in the list contains (a) an entry location, (b) an exit location, and (c) the timing information e.g., the date. The original Instagram check-in dataset also consists of a list. Each item in the list contains (a) a check-in location, (b) a comment from a user, and (c) the timing information of the check-in. Fig. 1 presents an illustration in which there are two items for the original flow dataset and two items for the original Instagram dataset.

We used the datasets that had been pre-processed. In other words, the flow dataset can be accessed as a set of flow data, and the Instagram check-in dataset can be accessed as a set of check-in data. Note that the locations with flow data and those with check-in data may differ (as indicated in Table I). In addition, our approach is capable of handling many other flow-based media datasets or check-in-based media datasets; it is not limited to the taxi dataset and the Instagram check-in dataset.

Based on the given datasets, this work will create a ranked list of events, where the ranked list can help people to quickly find the events that may be of interest. Table II illustrates the idea of a ranked list of two events in Sapporo. This list implies

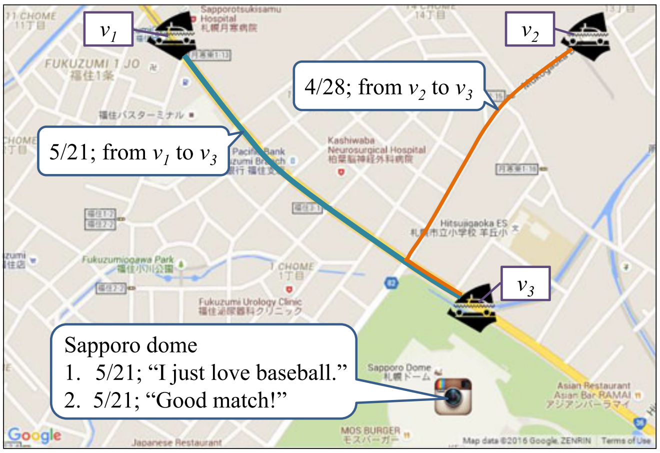
v_1
v_2
4/28; from v_2 to v_3
5/21; from v_1 to v_3
v_3
Sapporo dome
1. 5/21; "I just love baseball."
2. 5/21; "Good match!"
Sapporo Dome
Google
Map data ©2016 Google, ZENRIN

Fig. 1. Illustration of two items listed in the original flow dataset of taxis and two items in the original Instagram dataset that are associated with Sapporo Dome. The two datasets can be processed for a flow dataset with a set of flow data and a check-in dataset with a set of check-in data.

TABLE II
ILLUSTRATION OF A RANKED LIST OF EVENTS

Rank	Date	Location	Mined Essential Terms	Original Comments
1	9/20	Moerenuma Park	fireworks, art	*a.* wow! fireworks! *b.* Moerenuma art fireworks were great. ...
2	5/21	Sapporo Dome	baseball, match Nippon-Ham	*a.* I love baseball. *b.* Good match! *c.* Nippon-Ham is ... ...

Two items (i.e., rows) are listed, where each item is associated with an event. Each item provides the date, the location, the relevant terms, and users' comments that the relevant terms are derived from for the event.

that there could be two popular events: one is a display of fireworks in Moerenuma Park on September 20, and the other is a baseball game for the Nippon-Ham Fighters in Sapporo Dome on May 21. Additionally, more information may be obtained from the users' comments that are associated with the events. In the following, we will describe how to generate the ranked list of events.

IV. Proposed Approach

Fig. 2 outlines the proposed approach for finding events of interest in a specific region.

Given a set of flow data of locations and a set of check-in data of locations, the proposed approach normalizes the given sets of data, followed by combining the normalized datasets for a ranked list of events.

Specifically, we observed that the flow data of a location may vary over periods of time due to seasonal effects. For example, an increment in flow data can be observed during a tourist season. Therefore, the flow data of each location will first be processed by flow normalization, aiming to reduce the bias (Section IV-A). In contrast, the check-in data of each location will first be processed by buzz score calculation, aiming to identify critical terms from comments and/or tags of users (Section IV-B). The buzz score calculation can also be regarded as a type of normalization. Thus, the two datasets have been processed in similar ways. Subsequently, we calculate a score, called the variability score, for the data of each location and each date, which measures the degree of the variability of the data number for the location and the date for both datasets (Section IV-C). Calculating the scores is expected to *eliminate* the differences between the two given datasets. Thus, the scores from different datasets are comparable.

For data fusion, considering the geographic information is desirable because data for two locations might not be closely related if the locations are far away or have some other locations between them. We thus construct a graph called a spanning graph, which is capable of finding the critical neighborhood of given locations. The spanning graph will be used to link locations from different datasets (Section IV-D). We will fuse data only if their source locations are linked, i.e., they are considered to be closely related. We then assign weights for the edges of the graph, considering the distance information and the degree of connections (Section IV-E). Finally, the datasets will be fused by flow propagation from the flow dataset to the check-in dataset for a ranked list of events (Section IV-F).

A. Flow Normalization

For the taxi dataset, the numbers of entries and exits from taxis at a location change with the dates. Therefore, finding events of interest at the location should be achieved by observing the changes in such numbers for the location. However, we found that different periods of time may result in quite different numbers of entries and exits from taxis at a location. For example, we observed that the average numbers of entries and exits from taxis at the Sapporo station per day in August are typically greater than those in November, probably because of the tourist season in the City of Sapporo (August is tourist season, but November is not). With such a bias, it is likely that the most high-ranking events found by our approach will be in the tourist season or some other specific periods of time. (See Section IV-C for more details.)

Here, we intend to reduce the seasonal effect. We normalize the flow data of a date considering the relationship among the date and consecutive days around the date. Given the flow data of a location of a date, we would like to replace the flow data with the average of a list of ratios. Specifically, the ratios are obtained by dividing the flow data and each of a list of flow data of the same location for consecutive days around the given day. Formally, we normalize the flow data of a location of a date by

$$f_{i,j}^{N} = \frac{1}{2a+1} \sum_{s=j-a}^{j+a} \frac{f_{i,j}}{f_{i,s}} \tag{1}$$

where $f_{i,j}^{N}$ is the flow data of location i for date j after flow normalization, $f_{i,j}$ is the given flow data of location i for date j, and $a \geq 1$ is a user-defined parameter that specifies the data range used for the normalization. The summation is from a

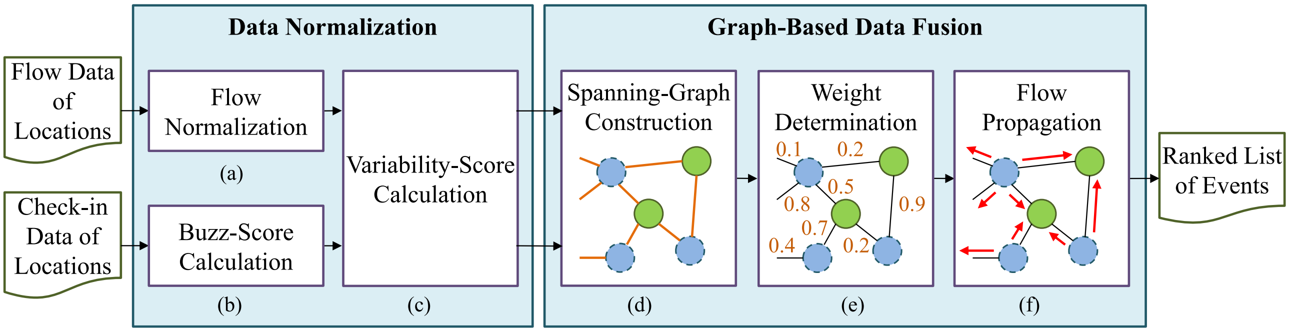

Fig. 2. Overall flow of the proposed approach for finding events of interest. The framework of the proposed approach has two stages: data normalization and graph-based data fusion. Note that data normalization is essential before data fusion because the given datasets are different in nature. Specifically, the approach uses six techniques, including (a) flow normalization, (b) buzz score calculation, (c) variability score calculation, (d) spanning graph construction, (e) weight determination, and (f) flow propagation. For these techniques, the first three are used for data normalization, and the other three are used for data fusion. Overall, the given datasets will be normalized individually and then combined using graph-based techniques.

days before date j to a days after date j. We can observe that the division for $f_{i,j}$ and $f_{i,s}$ helps us reduce the bias for a period of time. Empirically, a is set to 3; thus, the set of flow data for a week will be involved.

Practically, we will normalize the given set of flow data of each location for each date.

B. *Buzz Score Calculation*

Calculating buzz scores has recently been shown to be successful in trend searches based on query logs; see, e.g., [20], [21]. Given the query logs of a day, we can create a query set and calculate the frequency of the occurrence of each query. For trend searches, the basic idea is to calculate a score (i.e., buzz score) for each query of a day, where the score is the difference of the probability of using the query on the day and that of using the query on one day prior. In this way, the query that has the highest score implies that the query could be noticeable on the day. In this paper, we develop a strategy similar to that in [21] for trend searches on check-in data, where buzz score calculation helps us identify critical terms from the comments and/or tags of users.

Given all check-in data for each location, we first remove stop words that appear in the contents of tags and comments. We then split the contents of the tags and comments into terms. Given the terms, we extract the top frequent terms (top 100 terms in our work), and then we calculate their buzz scores individually. Assume that n terms, i.e., $\{t_1, t_2, ..., t_n\}$, have been extracted. As mentioned in [21], the buzz score of term t_k of a location of a date for $1 \leq k \leq n$ can be formulated as

$$c^B_{i,j,k} = \sum_{s=j-b}^{j-1} \left(\frac{1}{j-s} (P(t_k|T_j) - P(t_k|T_s)) \right) \tag{2}$$

where $c^B_{i,j,k}$ is the buzz score of term t_k of location i for date j, $b \geq 1$ is a user-defined parameter, and $P(t_k|T_j)$ is the probability of the occurrence of term t_k given the set of terms T_j for date j. The buzz score of a term for a date helps us measure the degree of increase in the popularity of the term on the date. A term with a high buzz score for a date implies that the frequency of using the term (mostly mentioned or discussed) has a large increase compared to a number of days before. $b \geq 1$ can be used to determine the number of days that will be involved before the date. As suggested in [21], we also reduce the effect for the dates far from date j based on $1/(j-s)$. Empirically, b is set to 5, where the effect of increasing the value of b is marginal.

Furthermore, we can consider groups of terms with similar semantics and then simply modify the formulation based on the same technique as that mentioned in [20], [21]. Assume that t_k and t_ℓ are groups of terms with similar semantics. The occurrence of t_ℓ will add a generalized count to t_k if t_ℓ includes t_k. For example, “Isaac Newton” and “Sir Isaac Newton” have similar semantics. The occurrence of “Sir Isaac Newton” will add a generalized count to “Isaac Newton.” The buzz score of t_k can then be modified as follows:

$$c'^B_{i,j,k} = c^B_{i,j,k} \times \log(1 + \mathrm{h}(t_k, i, j) + \mathrm{h}^*(t_k, i, j)) \tag{3}$$

where $\mathrm{h}(t_k, i, j)$ and $\mathrm{h}^*(t_k, i, j)$ are the count of the occurrence of t_k and the generalized count of t_k of location i for date j, respectively. Note that the modified buzz score of t_k will be greater than that of t_ℓ; thus, t_k will be more noticeable than t_ℓ.

Note that every location has its own top frequent terms. Practically, we will calculate the buzz score for each top frequent term of each location for each date. In addition, note that buzz score calculation and flow normalization are similar in concept because they both consider dates other than the date being processed. We did not use the same approach for the given datasets because their characteristics are different. However, we thought that calculating buzz scores can also be regarded as a type of normalization. Thus, we will not further normalize the check-in data prior to the variability score calculation.

C. *Variability Score Calculation*

Consider a period of time. Intuitively, finding events of interest should be achieved by observing the changes in data numbers along the dates. Whenever there is a large increase in data numbers on a date, there might be an event that occurred on that date. For example, consider a sequence of five numbers, say $\{10, 11, 20, 10, 11\}$. Assume that each number is associated with a date. It is likely that there was an event that occurred on the date associated with the third number in the sequence because there is a large increase from 11 to 20 on that date compared to the other dates.

Furthermore, we observed that it is preferred to observe the changes in data numbers along each of the seven days of a week (e.g., Sunday) rather than along the dates. The reason is that the change in data numbers may be considerable for different days of a week. For example, the numbers of entries and exits from taxis on weekdays are typically greater than those on holidays based on our dataset. The change along a day of a week is relatively regular; thus, observing the change is relatively effective. Consequently, we decompose the set of data for each location of the resulting dataset from Section IV-A into seven groups, which are associated with the seven days of a week. Similarly, we also decompose the set of data for each location of each top frequent term of the resulting dataset from Section IV-B into seven groups, which are associated with the seven days of a week. Eventually, for part of the flow-based dataset, each group is associated with a location and one of the seven days of a week. For part of the check-in-based dataset, each group is associated with a location, a term, and one of the seven days of a week.

Practically, we express each group as a sequence of numbers sorted along the dates. For example, consider a location and a period of time, e.g., a year. For the flow-based dataset, i.e., the taxi dataset, the sequence of Sunday will contain the number (after flow normalization) of entries and exits from taxis at the location for the first Sunday of the year, that for the second Sunday of the year, and so on. With all the sequences, we will observe the change in numbers in each sequence for the event of interest.

For each sequence, we first calculate the mean and the standard deviation. We then define the variability score of an entry of the sequence as the difference between the number associated with the entry and the mean divided by the standard deviation. The variability score associated with flow data is defined as

$$f^V_{i,j} = \frac{f^N_{i,j} - \mu^F_{i,s}}{\sigma^F_{i,s}}, s = \mathrm{d}(j) \tag{4}$$

where $f^N_{i,j}$ is the flow data of location i for date j after flow normalization, $f^V_{i,j}$ is the variability score of $f^N_{i,j}$, $\mathrm{d}(j)$ provides the day (e.g., Sunday) of the week for date j, and $\mu^F_{i,s}$ and $\sigma^F_{i,s}$ are the mean and the standard deviation, respectively, of the sequence associated with location i of day s of the week for the flow data. Note that the variability score can be either positive or negative depending on whether $f^N_{i,j}$ is greater than or less than $\mu^F_{i,s}$. A positive (negative) score implies that the statistic of flow is greater (less) than usual. The advantage of calculating variability scores is twofold.

1) Every variability score can be used to measure the degree of the variability of the data number for a specific location and a specific date. The larger the absolute value of a score is, the higher the variability of the data is. A positive (negative) score implies that the data number is greater (less) than normal.
2) Calculating variability scores can be regarded as a normalization that eliminates the unit of the given data numbers. It is beneficial for us to combine the flow-based dataset and the check-in-based dataset later.

Similarly, we can define the variability score associated with the check-in data as

$$c^V_{i,j,k} = \frac{c^B_{i,j,k} - \mu^C_{i,s,k}}{\sigma^C_{i,s,k}}, s = \mathrm{d}(j) \tag{5}$$

where $c^B_{i,j,k}$ is the buzz score of term t_k of location i for date j, $c^V_{i,j,k}$ is the variability score of $c^B_{i,j,k}$, $\mathrm{d}(j)$ indicates the day of the week for date j, and $\mu^C_{i,s,k}$ and $\sigma^C_{i,s,k}$ are the mean and the standard deviation, respectively, of the sequence associated with term t_k of location i of day s of the week for the check-in data.

As mentioned in Section IV-A, flow normalization is essential for reducing the bias from different periods of time. Without flow normalization, the variability scores of data in some period of time, e.g., a tourist season, may be considerably higher than those in the other periods of time such that most events with high rankings may be selected from that period of time (i.e., the tourist season). Similarly, the calculation of buzz scores also helps to reduce this type of bias.

D. Spanning Graph Construction

Thus far, the set of flow data and the set of check-in data have been normalized. We start to combine the normalized data from the given datasets. We consider that both of the datasets have location information, and there have been many studies on graph algorithms. Consequently, we intend to bridge the two datasets using graph-based techniques. Specifically, consider two types of vertices. Each of the locations in the set of flow data can be modeled as one type of vertex. Each of the locations in the set of check-in data can be modeled as the other type of vertex. Given two different types of vertices, we may add an edge for the two vertices if the locations associated with them are relevant. See Fig. 2(d) for an illustration. We then assign an appropriate weight for each edge, as shown in Fig. 2(e). Finally, we may combine the set of flow data and the set of check-in data by transmitting information along the edges, as shown in Fig. 2(f). This section will focus on graph construction, and Sections IV-E and IV-F will present weight determination and information transmission, respectively.

For graph construction, we implement spanning graphs [22], [23] for the two datasets. Note that using different graph construction methods, e.g., the k-nearest neighbors (k-NN), may still work. However, we believe that using spanning graphs is more suitable than using k-NN graphs in this work. (More details can be found in the last paragraph of this section.) To date, there have been many studies showing that constructing spanning graphs is capable of finding critical neighborhoods of vertices in a plane; see, e.g., [24]–[26].

A spanning graph is an undirected graph that contains no graph loops or multiple edges, and a spanning graph can simply be constructed as follows. Given a set of vertices in a plane, we may traverse the vertices in any order. Whenever a vertex is

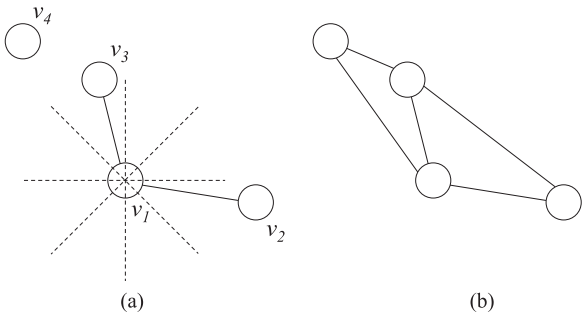

Fig. 3. Illustration of spanning graph construction. Consider a set of vertices: v_1, v_2, v_3, and v_4. (a) Assume that v_1 is visited first. We will evenly divide the plane into eight regions with respect to v_1. We then add an edge between v_1 and v_2 and an edge between v_1 and v_3 because each of v_2 and v_3 is the nearest vertex to v_1 in one of the regions. (b) The spanning graph for the given set of vertices.

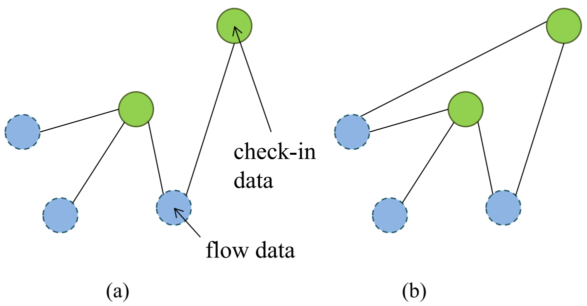

Fig. 4. Illustration of spanning graph construction in this work. Assume that we traverse vertices associated with the flow-based dataset, followed by those associated with the check-in-based dataset. Note that the resulting graph is independent of the traversed order. (a) The resulting graph after the vertices associated with the flow-based dataset have been traversed. (Assume that the vertices associated with the check-in-based dataset have not been traversed.) (b) The resulting spanning graph after all of the vertices have been traversed.

visited, we will divide the plane evenly into eight regions with respect to the vertex, followed by connecting the vertex to the nearest vertex in each region. Note that we may evenly divide the plane into more than eight regions if necessary. Eventually, we will have a spanning graph for the given set of vertices when all of the vertices have been visited. Overall, the number of edges (i.e., size) of a spanning graph is $O(n)$ if the number of vertices is n. Moreover, spanning graphs can efficiently be constructed [22]. Thus, constructing spanning graphs is capable of scaling to the case of a large number of vertices.

Fig. 3 shows an example of spanning graph construction for four vertices, i.e., v_1, v_2, v_3, and v_4.

These vertices are initially disconnected. Assume that v_1 is visited first. As shown in Fig. 3(a), v_1 is connected to v_2 and v_3 because each of v_2 and v_3 is the nearest vertex to v_1 in one of the eight regions with respect to v_1. Once all of the vertices have been visited, we will have a spanning graph for the four vertices, as shown in Fig. 3(b).

Note that we intend to combine two types of data numbers, i.e., flows and check-ins, not data numbers of the same type. We thus made some modifications here. We model each of the locations in the set of flow data as one type of vertex and each of the locations in the set of check-in data as another type of vertex. We then modify a part of the spanning graph construction method by restriction, as follows. Whenever a vertex v is visited, v will only consider those vertices of the other type with respect to v. In this way, each edge will only be used to connect vertices of different types. Eventually, we can obtain a graph for the two datasets (i.e., flow-based and check-in-based datasets). Fig. 4 presents an example.

Without loss of generality, assume that the vertices associated with the flow-based dataset will be traversed before those associated with the check-in-based dataset. Fig. 4(a) shows the resulting graph after the vertices associated with the flow-based dataset have been traversed, and Fig. 4(b) shows the resulting spanning graph after all of the vertices have been traversed.

Consider the process of connecting a vertex v for flow data to its nearest vertex for check-in data in each of the eight regions with respect to v. Interestingly, the process is somewhat similar to the case in which one may walk to a nearby location (in any direction) after exiting a taxi. From this perspective, we considered that using another method, e.g., k-NN graphs, for graph construction may work, but the resulting performance might not be better than that of using the spanning graphs (see Section V-D.2 for the evaluation).

E. Weight Determination

Given a graph, we intend to assign a weight for each edge. To further combine the two given datasets, we plan to transmit information from the vertices of the flow-based dataset to the vertices of the check-in-based dataset. Note that we do not consider the opposite direction for transmission. There are two reasons. The first reason is that the check-in-based dataset can offer textual information but our flow-based dataset cannot, where the textual information may tell us what occurred at the check-in locations. The second reason is that the information of event locations from the check-in-based dataset is typically more accurate than that from the flow-based dataset. In this way, the exact event locations will be determined from the locations in the check-in-based dataset.

For weight determination, we consider 1) how the statistics (i.e., variability scores) of a vertex for flow data are to be divided for its adjacent vertices, i.e., weights of distribution, and 2) the percentage of the input from a vertex v for flow data that a vertex for check-in data that is adjacent to v should receive, i.e., weights of collection. Accordingly, for each edge, we will assign two values; then, we set the product of the two values as the edge weight. Formally, given a graph, let E be the edge set of the graph. Let V^F and V^C be the vertex set for the flow-based dataset of the graph and the vertex set for the check-in-based dataset of the graph, respectively. For each edge, the first value (i.e., weight of distribution) can be set as

$$w^F_{p,q} = \frac{\left(\frac{1}{\mathrm{g}(p,q)}\right)^2}{\sum_{(p,r)\in E}\left(\frac{1}{\mathrm{g}(p,r)}\right)^2}, \quad \forall (p,q) \in E, p \in V^F, q \in V^C \tag{6}$$

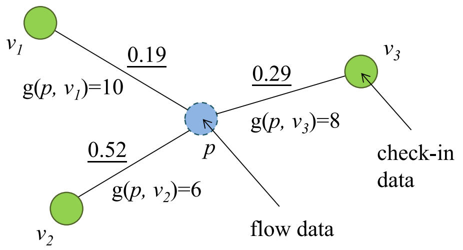

Fig. 5. Illustration of assigning values for edges (partial values for the desired weights) that are adjacent to a vertex for flow data. Assume that p is a vertex for flow data and v_1, v_2, and v_3 are the vertices for check-in data that are adjacent to p. If $\mathrm{g}(,)$ provides the geographical distance for two vertices, then the assigned values will be the values with underlines.

TABLE III
ILLUSTRATION OF AN ORIGINAL RANKED LIST OF EVENTS

Rank	Date	Location	Mined Essential Terms	Original Comments
1	9/20	Moerenuma Park	fireworks	wow! fireworks!
2	5/21	Sapporo Dome	baseball	I love baseball.
3	9/20	Moerenuma Park	art	Moerenuma art fireworks were great.
4	5/21	Sapporo Dome	match	Good match!
5	5/21	Sapporo Dome	Nippon-Ham	Nippon-Ham is the best∼

Items in the list will be merged together if they are for the same date and the same location. The merged (final) results of events of interest are presented in Table II.

where $w^F_{p,q}$ is the first value for the edge between vertex p and vertex q and $\mathrm{g}(p,q)$ provides the geographical distance between the two locations that are associated with p and q. An edge will be assigned a large value if both (a) the degree of its vertex for flow data and (b) the geographical distance associated with the edge are small compared to the other edges connected to the vertex for flow data. Fig. 5 presents an example, where p is a vertex with a degree of 3 for flow data and v_1, v_2, and v_3 are the vertices for check-in data that are adjacent to p. For each edge, the value with an underline shows the first value for the edge.

Similarly, the second value $w^C_{p,q}$ (i.e., weight of collection) can be set in a similar way such that an edge will be assigned a large value if both (a) the degree of its vertex for check-in data and (b) the geographical distance associated with the edge are small compared to the other edges connected to the vertex for the check-in data.

Finally, the overall weight of the edge between p and q will be set as the product of $w^F_{p,q}$ and $w^C_{p,q}$.

F. Flow Propagation

Thus far, we have 1) the variability score of flow data for each location for each date, 2) the variability score for check-in data for each term of each location for each date, and 3) a graph with weighted edges that connects the vertices associated with the locations of the flow-based dataset and the check-in-based dataset. We can now transmit information from the vertices of the flow-based dataset to the vertices of the check-in-based dataset along the weighted edges on the graph. In this way, events will be detected based not only on data changes for check-in locations but also on changes of traffic flows near these check-in locations.

For the check-in-based dataset, a score for each term of each location for each date can be determined using the following formula:

$$c^P_{q,j,k} = c^V_{q,j,k} \sum_{(p,q)\in E} \left(w^P_{p,q} \cdot f^V_{p,j} \right) \tag{7}$$

where $c^P_{q,j,k}$ is the resulting score for term t_k of location q for date j, $c^V_{q,j,k}$ is the variability score for the check-in data for term t_k of location q for date j, $w^P_{p,q}$ is the weight of the edge that connects location p and location q, and $f^V_{p,j}$ is the variability score for the flow data of location p for date j. Finally, we can generate a ranked list of events based on the resulting scores. The higher the score is, the higher the rank is. Note that $c^V_{q,j,k}$ or the summation of the products may be negative. We will remove any event data from the ranked list if both of the values associated with the event data are negative. The reason is that this typically occurs when more critical events occurred before or after the current date. Although the case of two negative values could imply an event occurring, it also implies that it is challenging to describe the event based on the textual information from the current datasets.

For the ranked list of events, we will merge the results for the same date and the same location, but with different terms, into a result that is with the union of the terms and with a higher rank. This is because we observed that these terms are likely to be associated with the same event. Consequently, we can use 1) a date, 2) a location (from the check-in-based dataset), and 3) one or more terms to describe each event in the ranked list. Practically, we can still efficiently extract comments that are related to the terms from users' comments for the location for the date if necessary.

Table III presents an example of a ranked list of events before merging its items. A total of five items (i.e., rows) are listed. The comments of each item are associated with the terms of the item (comments are not shown here due to the space limit for the table). Table II shows the resulting ranked list of events after merging the items in Table III that are with the same date and the same location. For example, fireworks are at Moerenuma Park on September 20.

V. EXPERIMENTAL RESULTS

A. Datasets

To evaluate the proposed approach, we conducted experiments based on a taxi dataset obtained by the cloud service Spatiowl [27] and an Instagram dataset for the City of Sapporo from March to November in 2014 (as mentioned in Section I).

The taxi dataset was generated based on the GPS data of the taxis, where GPS locations were grouped if their distance is within 50 meters. For the Instagram dataset, we used Instagram's open API [28] for the collection of check-in data. To use the API, we considered two issues. First, each request has an upper limit on the number of returned photos; thus, the returned result might be incomplete if a large search region is used. Second, we are also limited to 5,000 requests per hour per *access_token*, which makes using small search regions for the entire collection process impractical. Consequently, we followed a collection approach similar to that presented in [29] to collect as many photos as possible considering these issues. Specifically, we started from a large search region (i.e., a circle with a 5 km radius) and a time span (i.e., 1,000 seconds). Whenever the number of returned photos was greater than the limit, we divided the search region into four sub-regions and then called the search API recursively.

In total, the number of entry and exit locations is approximately 5,500 for the taxi dataset, and the number of check-in locations is approximately 6,300 for the Instagram dataset.

B. Experimental Setup

We will present 1) cross-dataset feature analysis and 2) comparative studies in Section V-D.1 and Section V-D, respectively. For the cross-dataset feature analysis, we focused on data from May to July for a detailed analysis. The results for the other months are similar. We studied (a) the normality, (b) the distribution, and (c) the cross-dataset coverage for the taxi dataset and the Instagram dataset. For comparative studies, we evaluated (a) the effect of using cross-domain social media (by comparing our approach with [21], i.e., a state-of-the-art approach), (b) the effect of using spanning graphs for connecting the two datasets, and (c) the effect of data normalization. We will show all the results ranging from March to November.

Note that all performances of the comparative studies were evaluated by P@n, which represents the precision (i.e., correctness) of the top n events in the ranking results. Because there is no ground truth in the media datasets, we manually pre-defined a total of 212 events from the nine months, and then we used these events as our ground truth. These events are associated with sports, local festivals, concerts, and exhibitions. In particular, Sapporo has a soccer team and a baseball team. There are many sport events that occur every year. Moreover, Sapporo has many seasonal local festivals, e.g., beer festival and snow festival. Therefore, more than 70% of the pre-defined events are either sports or local festivals.

C. Cross-Dataset Feature Analysis

Regarding the two datasets, this section will analyze the normality using the Shapiro-Wilk test [30], compare the distributions of their popular locations, and explore the cross-dataset coverage.

1) Normality Analysis: We mentioned in Section IV-C that it is preferred to observe the changes of data numbers along each of the seven days of a week rather than along the dates. We intend to show that data numbers along the days of a week are more likely to be regular than those along the dates using normality tests. Specifically, we intend to verify that data numbers along the days of a week are more likely to be normally distributed than those along the dates.

TABLE IV
RESULTS OF THE SHAPIRO-WILK NORMALITY TEST ON THE DATA NUMBERS ALONG THE DATES

Month(s)	p-value	
	Taxis	Instagram
May	0.006985	0.000182
June	0.017530	0.000611
July	0.001193	0.066980
May–July	0.000012	0.000010

From the p-values, we can conclude that all the data populations are not normally distributed, except for the data population of Instagram's data numbers in July.

TABLE V
RESULTS OF THE SHAPIRO-WILK NORMALITY TEST ON THE DATA NUMBERS ALONG THE DAYS OF A WEEK FOR MAY TO JULY

Day	p-value	
	Taxis	Instagram
Sunday	0.017910	0.664800
Monday	0.010630	0.010370
Tuesday	0.000320	0.134700
Wednesday	0.754200	0.013480
Thursday	0.488900	0.049050
Friday	0.013780	0.605100
Saturday	0.997300	0.644700

From the p-values, we can conclude that half of the data populations are not normally distributed.

For each dataset, we simply summed up all data numbers in a day for all locations. In this way, we obtained a data number for each date for each dataset. For normality analysis, we used the Shapiro-Wilk test [30], where the null hypothesis of the test is that a population is normally distributed. The p-value obtained from the test shows the probability that the given data numbers are from a normal distribution. The lower the p-value is, the smaller the probability is. Because many statisticians use a value of 0.05 as a threshold, we will reject the null hypothesis if the p-value is less than 0.05.

Table IV shows the p-values for the data numbers along the date based on the Shapiro-Wilk test, where data numbers of "May–July" were obtained from concatenating data numbers for the three months.

As shown, except for Instagram's data population in July, it is very likely that the other populations were not from a normally distributed population. In contrast, Table V shows the p-values for the data numbers along the days of a week for May to July based on the Shapiro-Wilk test.

As shown, it is likely that only half of the data populations were not from normally distributed populations.

Comparing Tables IV and V, we concluded that data numbers along the seven days of a week are more likely to be regular than those along the dates.

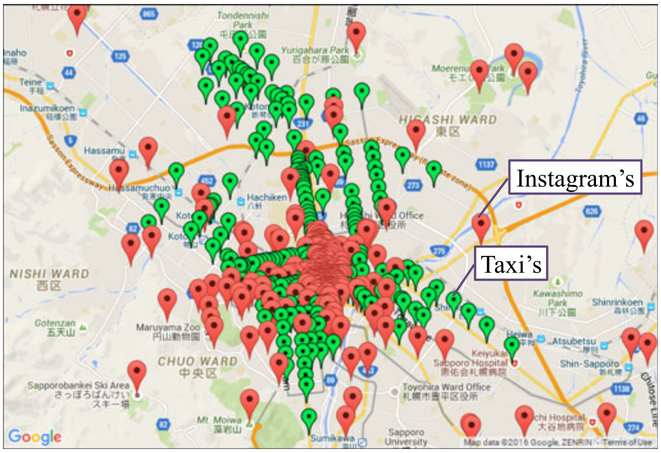

Fig. 6. Distribution of 1) the top 300 popular entry and exit locations (green) for the taxi dataset and those of 2) the top 300 popular check-in locations (red) for the Instagram dataset in the city of Sapporo. The popular locations for the taxi dataset are relatively clustered, whereas those for the Instagram dataset are relatively scattered.

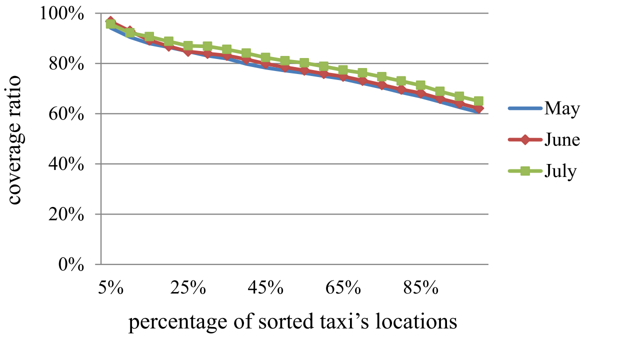

Fig. 7. Change of coverage ratios from the Instagram dataset along with changes in the percentages of entry and exit locations for the taxi dataset based on popularity. Most of the popular taxi locations are close to some Instagram locations, but the ratio decreases as increasingly more non-popular taxi locations are involved.

2) Distribution Analysis: Consider the entry and exit locations of the taxi dataset and the check-in locations of the Instagram dataset. Fig. 6 shows the top 300 popular locations for the taxi dataset and those for the Instagram dataset in the City of Sapporo.

We can observe that the majority of the popular locations in the two datasets are clustered in the city center, but overall, their distributions are slightly different. The popular locations in the taxi dataset are relatively clustered. Many of the locations appear along some major roads or near the intersection of roads. In contrast, the popular locations in the Instagram dataset are relatively scattered. Many of the locations are tourist attractions. Note that the attractions may also be entry or exit locations for the taxi dataset (but not any of the top 300 popular locations). Similarly, popular locations for the taxi dataset may also be check-in locations for the Instagram dataset. From this perspective, the two datasets may compensate each other.

3) Coverage Analysis: We evaluate the correlation between the taxi dataset and the Instagram dataset through coverage analysis. We consider the entry and exit locations for the taxi dataset and the check-in locations for the Instagram dataset. For each dataset, we sorted the locations according to the degree of popularity, from popular locations to non-popular locations. For the taxi dataset, the popularity of a location is determined by the total number of people who enter or exit taxis at the location. The larger the number is, the more popular the location is. For the Instagram dataset, the popularity of a location is determined by the total number of check-ins for the location. Similarly, a location with a large number of check-ins implies that the location is popular. We then find the ratio of the locations from one dataset that are close to (say, within 100 meters) the locations from the other dataset considering the degree of popularity.

Fig. 7 shows the ratios of the locations for the taxi dataset that are close to the locations for the Instagram dataset, where the numbers listed on the horizontal axis represent the percentages of the entry and exit locations for the taxi dataset based on popularity. For example, 5% denotes the top 5% (approximately 275) popular locations. The coverage ratio of the top 5% taxi locations is approximately 94%, which means that 94% of the taxi locations are close to some Instagram locations.

Overall, the change in the coverage ratios implies that most of the popular taxi locations are close to some Instagram locations, but the ratio decreases as increasingly more non-popular taxi locations are involved. In other words, there might be no Instagram locations that are close to the non-popular taxi locations.

Similarly, we also observed the ratios of the locations for the Instagram dataset that are close to the locations for the taxi dataset, and the trend is similar to that in Fig. 7. Overall, approximately 74% of the top 5% (approximately 315) popular Instagram locations are close to some taxi locations, and the ratio decreases as increasingly more non-popular Instagram locations are involved. We observed that the coverage ratio, 74%, is smaller than the coverage ratio, 94% (see Fig. 7). Consistent with Fig. 6, this result may be because the distribution of the popular Instagram locations is more scattered than that of the popular taxi locations. (Note that the coverage ratios shown in Table I are average values.)

D. Evaluation of the Proposed Approach

In the following, we evaluate the accuracy of detected events for the proposed approaches. Section V-D.1 shows the effect of cross-domain media mining, Section V-D.2 shows the effect of spanning graph construction, and finally, Section V-D.3 shows the effect of data normalization.

1) Effect of Cross-Domain Media Mining: Fig. 8 compares the precision of detecting events when using the Instagram dataset alone and the results of using both the taxi dataset and the Instagram dataset, where the approach of using the Instagram dataset alone was implemented based on that in [21].

As shown, combining the two datasets achieved better performance for finding events in the nine months. Based on the results, we found that some events can only be detected by using the Instagram dataset alone. However, the time information for the events might not be sufficiently accurate. The reason is that people might not upload their media data for an event immediately but rather a few days after the event. Perhaps some of

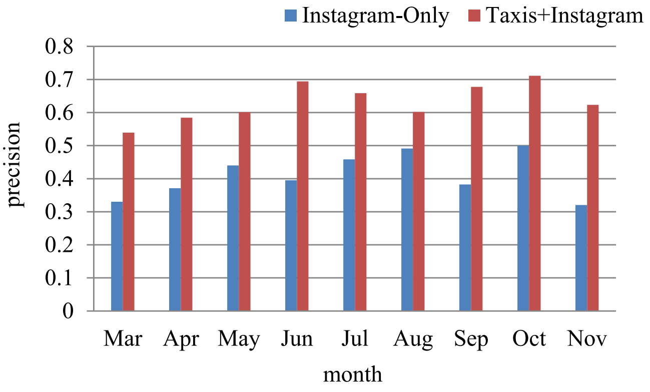

Fig. 8. Comparison of the detected results of using the Instagram dataset alone and the results of using both the taxi dataset and the Instagram dataset.

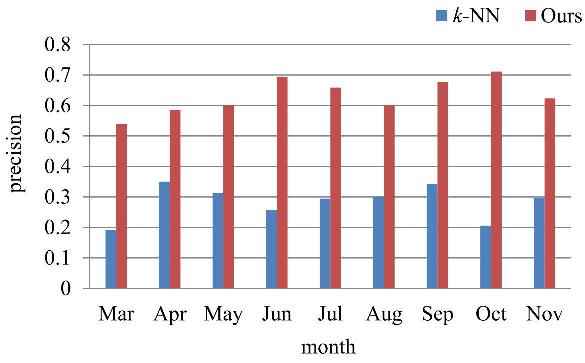

Fig. 9. Comparison of the results of using the k-nearest neighbors (k-NN) algorithm for graph construction and the results of using spanning graphs (see Section IV-D) for our work.

TABLE VI
COMPARISON OF THE RESULTS OF NO DATA NORMALIZATION AND THE RESULTS OF USING DATA NORMALIZATION

	Ours w/o Normalization			
	P@1	P@2	P@5	P@10
Pre-defined 10	0.1814	0.1925	0.2144	0.2177
Pre-defined 20	0.1942	0.2013	0.2472	0.2528
Pre-defined 50	0.2004	0.2391	0.2529	0.2643
All pre-defined	0.2243	0.2563	0.3074	0.3112
	Ours			
	P@1	P@2	P@5	P@10
Pre-defined 10	0.4292	0.5452	0.5727	0.6047
Pre-defined 20	0.4385	0.5561	0.5733	0.6126
Pre-defined 50	0.5348	0.5727	0.6527	0.6166
All pre-defined	0.5575	0.5796	0.6629	0.6387

Note that "Ours w/o Normalization" did not use flow normalization and variability score calculation, but it did use buzz score calculation. Every value is an average of values for nine months.

them like to remove some photos, add tags for someone, or make some comments. In contrast, the information provided by the taxi dataset may be more real time (as summarized in Table I). Adding the taxi dataset is thus beneficial for good performance.

2) Effect of Spanning Graph Construction: We conducted experiments on the use of the k-NN for graph construction, where the k-NN is a classical algorithm that generally works well in practice. For graph construction, the k-NN algorithm will connect each vertex to its nearest k vertices. In contrast, for spanning graph construction, given a vertex, the vertex connects only the nearest vertex in each of the eight regions with respect to itself (see Section IV-D). Fig. 9 compares the results of using the k-NN algorithm for graph construction and the results of using spanning graphs, where the k-value of the k-NN algorithm was set to 8 because the spanning graphs used in our work consider only eight regions.

(The construction of spanning graphs can easily be extended to consider more than eight regions.) Based on the results, the use of spanning graphs achieved better performance. We found that k-NN may make a vertex connect to many vertices that are in similar directions to the vertex, and generally, this situation will mislead the information transmission along the edges.

3) Effect of Data Normalization: Finally, this section evaluates the effect of data normalization (see Sections IV-A to IV-C). Specifically, we intend to evaluate the effect of the flow normalization and the variability score calculation, except for the buzz score calculation. Buzz score calculation is essential for us to extract critical textual information. We consider the correctness, i.e., P@n, for the top 10 events (pre-defined 10), top 20 events, top 50 events, and all events (*i.e*, 212 events), where the top 10 events refer to the 10 events that have the most data volume among all the events, and so forth. Table VI compares the results of no normalization (i.e., no flow normalization and no variability score calculation) and the results of using data normalization.

As expected, normalization does play an important role in combining datasets with different *units*, e.g., the "times" of entering and exiting taxis, for high performance. We found that data from the taxi dataset dominated the ranking results if no normalization was involved. This result is because the data volume of the taxi dataset is considerably greater than that of the textual data from the Instagram dataset in our work.

VI. CONCLUSION

This paper has presented a two-stage framework that includes data normalization and graph-based data fusion for unifying a flow-based dataset and a check-in-based dataset for automatic event discovery. Based on a taxi dataset and an Instagram dataset, the experiments have shown that combining the two datasets can achieve a 57% precision improvement on average compared with a state-of-the-art approach of using the Instagram dataset alone. Future works include the fusion of more than two datasets and the exploration of image data for better performance. A simple extension of the framework for more than one dataset is described as follows.

If there are several flow-based datasets and several check-in-based datasets, a simple approach may first run the first stage of the framework for each dataset. Because the calculation of variability scores makes different datasets comparable, we can consider all flow-based datasets (check-in-based datasets) as one flow-based dataset (check-in-based dataset). Then, we can still run the second stage of the framework for events of interest. If there are several datasets and all of them are of the same type, a simple approach may still run the first stage of the framework for each dataset. Our approach for spanning graph construction will directly be extended such that each edge will only be used to connect vertices associated with different datasets. The current approach for weight determination is still used. The main difference is that the process of propagation now becomes propagating data from all vertices. If all datasets are flow-based (check-in-based) datasets, then the process becomes refining the variability scores of the flow data (check-in data). Comparing the variability scores, we may find a ranked list of events. Note that we will lose the textual information if all datasets are flow-based datasets.

Considering image data, we may generate image tags using existing annotation approaches if only a few images are tagged. The images can then be regarded as users' comments for the proposed framework. Meanwhile, images may be used to remove noisy terms from users' comments by computing term relevance for given images. In addition, in the future, we may discover relations between events and traffic jams for traffic route recommendations.

Acknowledgment

The authors would like to thank Prof. K. Aizawa of the University of Tokyo for valuable discussions, Y.-H. Kuo and W.-F. Cheng of National Taiwan University for valuable technical assistance, and to the anonymous reviewers for their helpful comments.

References

[1] Facebook, "Form 10-K (annual report)–Filed 02/01/13 for the period ending 12/31/12," 2013.

[2] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes Twitter users: Real-time event detection by social sensors," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 851–860.

[3] Y.-H. Kuo *et al.*, "Discovering the city by mining diverse and multimodal data streams," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 201–204.

[4] H. Becker, D. Iter, M. Naaman, and L. Gravano, "Identifying content for planned events across social media sites," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 533–542.

[5] W.-Y. Lee, Y.-H. Kuo, W. H. Hsu, and K. Aizawa, "City-view image location identification by multiple geo-social media and graph-based image cluster refinement," *J. Vis. Commun. Image Represent.*, vol. 41, pp. 200–211, 2016.

[6] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling, "TwitterStand: News in tweets," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2009, pp. 42–51.

[7] L. M. Aiello *et al.*, "Sensing trending topics in Twitter," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1268–1282, Oct. 2013.

[8] J. Weng, Y. Yao, E. Leonardi, and B.-S. Lee, "Event detection in Twitter," HP Laboratories: Singapore, Tech. Rep. HPL-2011-98, 2011.

[9] P. Meladianos, G. Nikolentzos, F. Rousseau, Y. Stavrakas, and M. Vazirgiannis, "Degeneracy-based real-time sub-event detection in Twitter stream," in *Proc. 9th Int. AAAI Conf. Web Social Media*, 2015, pp. 248–257.

[10] M.-S. Dao, D.-T. Dang-Nguyen, and F. G. D. Natale, "Robust event discovery from photo collections using signature image bases (SIBs)," *Multimedia Tools Appl.*, vol. 70, no. 1, pp. 25–53, 2014.

[11] M. Ruocco and H. Ramampiaro, "A scalable algorithm for extraction and clustering of event-related pictures," *Multimedia Tools Appl.*, vol. 70, no. 1, pp. 55–88, 2014.

[12] R. R. Shah *et al.*, "Concept-level multimodal ranking of Flickr photo tags via recall based weighting," in *Proc. ACM Workshop Multimedia COMMONS*, 2016, pp. 19–26.

[13] Z. Ma, Y. Yang, N. Sebe, K. Zheng, and A. G. Hauptmann, "Multimedia event detection using a classifier-specific intermediate representation," *IEEE Trans. Multimedia*, vol. 15, no. 7, pp. 1628–1637, Nov. 2013.

[14] Y. Yan *et al.*, "Complex event detection via event oriented dictionary learning," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 3841–3847.

[15] X. Chang, Y. Yang, A. G. Hauptmann, E. P. Xing, and Y.-L. Yu, "Semantic concept discovery for large-scale zero-shot event detection," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 2234–2240.

[16] M. Mazloom, X. Li, and C. G. M. Snoek, "TagBook: A semantic video representation without supervision for event detection," *IEEE Trans. Multimedia*, vol. 18, no. 7, pp. 1378–1388, Jul. 2016.

[17] W. Zhang *et al.*, "City-scale social event detection and evaluation with taxi traces," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 40:1–40:20, 2015.

[18] R. R. Shah *et al.*, "EventBuilder: real-time multimedia event summarization by visualizing social media," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 185–188.

[19] R. R. Shah, Y. Yu, and R. Zimmermann, "ADVISOR: Personalized video soundtrack recommendation by late fusion with heuristic rankings," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 607–616.

[20] Z. Al Bawab, G. H. Mills, and J.-F. Crespo, "Finding trending local topics in search queries for personalization of a recommendation system," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl, Discovery Data Mining*, 2012, pp. 397–405.

[21] C.-C. Wu, T. Mei, W. H. Hsu, and Y. Rui, "Learning to personalize trending image search suggestion," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2014, pp. 727–736.

[22] A. C.-C. Yao, "On constructing minimum spanning trees in k-dimensional spaces and related problems," *SIAM J. Comput.*, vol. 11, no. 4, pp. 721–736, 1982.

[23] H. Zhou, N. Shenoy, and W. Nicholls, "Efficient minimum spanning tree construction without Delaunay triangulation," *Inform. Process. Lett.*, vol. 81, no. 5, pp. 271–276, 2002.

[24] K.-H. Ho, H.-C. Ou, Y.-W. Chang, and H.-F. Tsao, "Coupling-aware length-ratio-matching routing for capacitor arrays in analog integrated circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 2, pp. 161–172, Feb. 2015.

[25] S.-L. Huang, C.-A. Wu, K.-F. Tang, C.-H. Hsu, and C.-Y. R. Huang, "A robust ECO engine by resource-constraint-aware technology mapping and incremental routing optimization," in *Proc. 16th Asia South Pacific Des. Autom. Conf.*, 2011, pp. 382–387.

[26] J. Long, H. Zhou, and S. O. Memik, "An $O(n \log n)$ edge-based algorithm for obstacle-avoiding rectilinear Steiner tree construction," in *Proc. Int. Symp. Physical Des.*, 2008, pp. 126–133.

[27] "Spatiowl," 2017. [Online]. Available: http://www.fujitsu.com/global/solutions/business-technology/intelligent-society/smart-mobility/spatiowl/

[28] "API endpoints," 2015. [Online]. Available: http://instagram.com/developer/endpoints/media/

[29] T. Fujisaka, R. Lee, and K. Sumiya, "Discovery of user behavior patterns from geo-tagged micro-blogs," in *Proc. 4th Int. Conf. Uniquitous Inf. Manage. Commun.*, 2010, pp. 246–255.

[30] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *OUP Biometrika*, vol. 52, no. 3–4, pp. 591–611, 1965.

Wen-Yu Lee received the Ph.D. degree in computer science from National Taiwan University, Taipei, Taiwan, in 2016.

She is currently a Postdoctoral Researcher with the University of Tokyo, Tokyo, Japan. Her current research interests include multimedia content analysis, image retrieval, and mining over large-scale databases.

Winston H. Hsu (S'03–M'07–SM'12) received the Ph.D. degree in electrical engineering from Columbia University, New York, NY, USA.

He previously worked in the multimedia software industry. Since 2007, he has been a Professor with the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan. He is also with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. His research interests include computer vision, machine intelligence, image/video indexing and retrieval, and mining over large-scale databases.

Dr. Hsu serves as the Associate Editor for the IEEE Transactions on Multimedia and on the Editorial Board for the *IEEE MultiMedia Magazine*.

Shin'ichi Satoh (A'11–M'17) received the B.E., M.E., and Ph.D. degrees from the University of Tokyo, Tokyo, Japan, in 1987, 1989, and 1992, respectively.

He is currently a Professor with the National Institute of Informatics, Tokyo, Japan. He was a Visiting Scientist with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, from 1995 to 1997. His research interests include video analysis and multimedia databases.

Dr. Satoh is a Member of the IPSJ, the ITEJ, the IEEE Computer Society, and the ACM.