Visual Event Mining from the Twitter Stream

Takamu Kaneko Keiji Yanai

Department of Informatics, The University of Electro-Communications, Tokyo
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585 JAPAN

1. INTRODUCTION

Twitter is a unique microblog, which is different from conventional social media in terms of its quickness and on-the-spot-ness. Many Twitter's users send messages, which is commonly called "tweets", to Twitter on the spot with mobile phones or smart phones, and some of them send photos and geotags as well as tweets. Most of the photos are sent to Twitter soon after taken. In case of photos related to some events, most of them are taken during the events. We think that Twitter event photo mining is more useful to understand what happens currently over the world than only text-based Twitter event mining.

In this paper, we propose a system to mine events visually from the Twitter stream. To do that, we use not only tweets having both geotags and photos but also tweets having geotags or photos for textual analysis or visual analysis.

Although there exist many works related to Twitter mining using only text analysis such as typhoon and earthquake detection by Sakaki et al. [1], only a limited number of works exist on Twitter mining using image analysis. Nakaji et al. [2] proposed a system to mine representative photos related to the given keyword or term from a large number of geo-tweet photos. They extracted representative photos related to events such as "typhoon" and "New Year's Day". They used only geotagged photo tweets the number of which are limited compared to all the photo tweets. Gao et al. [3] proposed a method to mine brand product photos from Weibo which employs supervised image recognition, which is different from event detection. They integrated visual features and social factors (users, relations, and locations) as well as textual features for brand product photo mining.

In this paper, we detect visual events using geotagged non-photo tweets and non-geotagged photo tweets as well as geotagged photo tweets. In the experiments, we show some examples of detected events and their photos such as "rainbow", "fireworks" and "festival".

2. VISUAL EVENT DETECTION

In this section, we overview a system to mine events from the Twitter stream. We propose a Twitter visual event mining system which consists of event keyword detection, location estimation of non-geotagged photos, event photo clustering, and representative photo selection.

The input data of the system are the tweets having geotags or photos (geo-tweets or photo tweets) gathered via the Twitter streaming API. We use geotagged tweets for event word detection, and photo tweets for event photo detection. The output of the system are event sets consisting of event words, geo-locations, event date, representative photos, and event photo sets. The system has GUI which shows detected events on the online maps.

The processing flow of the new system is as follows:

(1) Calculate area weights and "commonness score" of words in advance.

(2) Detect event word bursts using N-gram from geotagged tweets

(3) Estimate locations of non-geotagged photos

(4) Select photos and representative photos corresponding to the detected events

(5) Show the detected events with their representative photos on the map (See Fig.1 and Fig.2)

2.1 Textual Analysis

To detect events, we search for bursting keywords by examining difference between the daily frequency and the average daily frequency over a month within each unit area. The area which is a location unit to detect events is defined with a grid of 0.5 degree latitude height and 0.5 degree longitude width. In case that the daily frequency of the specific keyword within one grid area increases greatly compared to the average frequency, we consider that an event related to the specific keyword happened within the area in that day.

To detect bursting keywords, we calculate an adjusting weight, $W_{i,j}$, regarding the number of Twitter unique users in a grid, and a "commonness score ", $Com(w)$, of a word over all the target area in advance. To boost the areas with low activity and handle all the areas equally in the burst keyword detection, we introduce $W_{i,j}$ representing a weight to adjust the scale of the number of daily tweet users, which is defined in the following equation:

$$W_{i,j} = \frac{\#users_{max} + s}{\#users_{i,j} + s}, \quad (1)$$

where i, j, $\#users_{i,j}$, $\#users_{max}$ and s represents the index of grids, the number of unique users in the given grid, the maximum number of unique users among all the grids, and the standard deviation of user number over all the grids, respectively.

Next, we prepare a "commonness score" of each of the word appearing in Tweet messages by the following equation:

$$Com(w) = \sum_{i,j} \frac{E(\#users_{w,i,j})^2}{V(\#users_{w,i,j}) + 1}, \quad (2)$$

where i, j, $E(\#users_{w,i,j})$ and $V(\#users_{w,i,j})$ represents the index of grids, and the average number and the variance value of unique users who tweeted messages containing the given word w in the given grid in a day, respectively. The "commonness score" is used as a standard value for word burst detection.

In this paper, we use N-gram to detect burst words which does not need word dictionaries. As a unit of N-Gram, we use a character in Japanese texts and a word in English texts. First we count the number of unique users who posted Twitter messages including each unit within each location grid. We merge adjacent units both of which are contained in the messages tweeted by more than five unique users one after another.

$$S_{w,i,j} = \frac{\#users_{w,i,j}}{Com(w)} W_{i,j}, \quad (3)$$

where $\#users_{w,i,j}$ is the number of the unique users who tweeted messages containing w in the location grid (i, j). A word burst score, S, represents the extent of burst of the given word taking account of an area weight of the given location grid, $W_{i,j}$, and a "commonness score" of the given word, $Com(w)$. We regard the word the burst score of which exceeds the pre-defined threshold. In the experiments for Japan tweets, we set the threshold as 200. Note that when multiple words which overlap with each other are detected as events, we merge them into one event word.

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2889418 .

2.2 Location Estimation for non-geotagged photos

The photos embedded in the geotagged tweets from the messages of which the event words were detected in the given day and the given area can be regarded as event photos corresponding to the detected event. In this step, by using them as training data, we detect additional event photos from the non-geotagged photo tweets posted in the same time period as the detected event words. As a method, we adopt two-class classification to judge if each tweet photo corresponds to the given event or not.

To classify non-geotagged tweet photos into event photos or non-event photos, we propose a hybrid method of text-based Naive Bayes (NB) classifier and image-based Naive Bayes Nearest Neighbor (NBNN) [4]. We use Naive Bayes which is a well-known method for text classification to classify tweet messages, and NBNN which is local-feature-based method for image classification to classify tweet photos.

We use message texts and photos of geotagged tweets where the given event word are extracted as positive samples, and message texts and photos of geotagged tweets which include the given event words but were posted from the other areas as negative samples. For NB, we count the word frequency in positive and negative samples, while for NBNN, we extract SIFT features from sample images. To classify photos in the same way as NB, we use a cosine similarity between L2-normalized SIFT features instead of Euclid distance used in the normal NBNN.

The equation to judge if the given non-geotagged tweet photo corresponds to the given event or not is as follows:

$$\hat{c} = \arg\max_{c} P(c) \prod_{i=1}^{n} P(x_i|c) \sum_{j=1}^{v} \frac{d_j \cdot NN_c(d_j)}{\|d_j\| \|NN_c(d_j)\|}, \quad (4)$$

where n, x_i, v, d_j, and $NN_c(d_j)$ represents the number of words in the given tweet, the i-th words, the number of extracted local features from the photo of the given tweet, local feature vectors of SIFT, and the nearest local feature vectors of d_j in the training sample of class c which corresponds to "positive" or "negative", respectively.

2.3 Visual Analysis

Until the previous step, event keywords and their corresponding tweets have been selected. In this step, we carry out clustering and representative photo selection for the photos embedded in the selected event tweets and the photos selected from the non-geotagged photo tweets in the previous step.

As image features, we use an activation feature extracted from Deep Convolutional Neural Network (DCNN) pre-trained with ImageNet 1000 categories. We extract 4096-dim L2-normalized DCNN features using Overfeat [5] as a feature extractor.

For clustering photos, we use the Ward method which is one of agglomerative hierarchical clustering methods. It creates clusters so to minimize the total distance between the center of each cluster and the cluster members. It merges the cluster pairs which bring the minimum total error calculated in the following equation one by one.

We evaluate each of the obtained clusters in terms of visual coherence. We calculate visual coherence score V_C. When V_C is high, the corresponding cluster is likely to strongly related to the event. On the other hand, in case that V_C is lower, the cluster is expected to be a noise one which is less related to the event.

In addition, the cluster having the maximum value of V_C is regarded as a representative cluster, and the photo the visual feature vector of which is the closest to the cluster center is selected as a representative photo for the corresponding event.

3. EXPERIMENTAL RESULTS

We used the tweet data which was collected in August 2012. The number of geotagged photo tweets, geotagged non-photo tweets and non-geotagged photo tweets we collected in August 2012 were 255,455, 2,102,151 and 3,367,169, respectively. In advance, we calculated area weights and commonness score of words using all

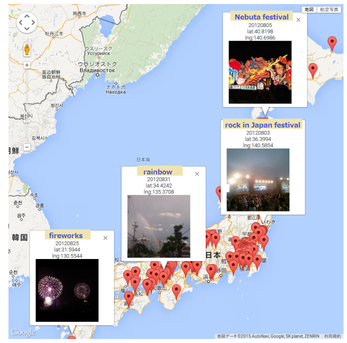

Figure 1: Example of detected events shown on the online map.

Figure 2: "Fireworks festival" photos automatically detected by the proposed system.

Table 1: Part of the detected events.

event name	date	lat,lng	Event Score	# photos	# photos (BL)
fireworks	2012/08/01	33,129.5	297.7	38	10
rainbow	2012/08/01	34,134.5	229.1	21	18
ROCK IN JAPAN	2012/08/03	36,140	430.3	51	not detected
Ayu Festival	2012/08/04	34.5,138.5	265.1	28	not detected
Nebuta Festival	2012/08/06	40.5,140	255.7	37	not detected
Awa-odori	2012/08/14	34,134	589.8	31	16
lightning	2012/08/18	34,135	367.5	106	37
blue moon	2012/08/31	34.5,136	269.7	69	59

the geotagged tweets. For comparison, we prepare a baseline system which uses only geo-tagged photo tweets.

The proposed system detected 310 events, while the baseline system using only geotagged photo tweets detected only 35 events which were about one ninth times as many as the proposed system.

Tab.1 shows parts of detected events including event names, location, date and event scores. 8 events shown in the table were detected by the proposed system, while the baseline system using only geotagged photo tweets detected only 5 out of 8. Regarding the number of detected photos, it was increased compared to the baseline (BL).

Some detected events are shown on the map with their representative photos in Fig.1. These map are interactive maps based on Google Maps API, and a user can see any event photos by clicking markers on the maps. Fig.2 shows detected "Fireworks festival" photos after clicking the representative photo shown in the pop-up maker.

4. REFERENCES

[1] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proc. of the International World Wide Web Conference*, pp. 851–860, 2010.

[2] Y. Nakaji and K. Yanai, "Visualization of real world events with geotagged tweet photos," in *Proc. of IEEE ICME Workshop on Social Media Computing (SMC)*, 2012.

[3] Y. Gao, F. Wang, H. Luan, and T.-S. Chua, "Brand data gathering from live social media streams," in *Proc. of ACM International Conference on Multimedia Retrieval*, 2014.

[4] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proc. of IEEE Computer Vision and Pattern Recognition*, 2008.

[5] P. Sermanet, D. Eigen, X. Zhang, M.l Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *Proc. of International Conference on Learning Representations*, 2014.