Wisdom of the Local Crowd: Detecting Local Events Using Social Media Data

Søren B. Ranneries[1] Mads E. Kalør[2] Sofie Aa. Nielsen[3]
Lukas N. Dalgaard[4] Lasse D. Christensen[5] Nattiya Kanhabua[6]
Department of Computer Science, Aalborg University, Denmark
{[1]sranne12, [2]mkalar12, [3]sani12, [4]lnda12, [5]ldch11}@student.aau.dk, [6]nattiya@cs.aau.dk

ABSTRACT

Event attendees post about their experiences on social media. We propose a novel approach for analyzing these posts to extract ongoing events. We gather posts from Twitter and Instagram and perform a number of processing steps to identify posts related to events based on hashtags and location information. Our approach detects events only using posts submitted during the past hour which ensures that only ongoing events are detected. The system can detect both large and small events with a high location accuracy, a precision of 0.20, and a recall of 0.60.

CCS Concepts

•Information systems → Location based services; Information extraction;

Keywords

Local Event Detection, Social Media, Twitter, Instagram

1. INTRODUCTION

The desire to go out often arises spontaneously. If nothing is planned one must figure out what is happening now or in the immediate future. There are several ways to find out what is going on. One can ask friends, check the calendar of venues, or look at event aggregation sites — but no single source has an exhaustive list of events happening right now or nearby. This paper attempts to identify local events by analyzing posts on social media. However, extracting information about events from posts generated by the users of social media is a challenging task as the data is noisy and unstructured. In addition, the sheer volume of data generated demands for an automated approach to solving this challenge.

Other studies indicate that social media data can be used for event detection. Lee and Sumiya [3] present a method for detecting events by irregularities in the rate of posts, unique users, and the movement of users. However, their approach is unable to detect small-scale events such as performances by street musicians. An approach by Walther and Kaisser [6] uses a location based clustering to detect events. However, their approach does not account for several events happening in the same area. We attempt to solve this shortcoming by performing both location and topical clustering. To the best of our knowledge, all previous work is solely based on Twitter data. A study shows that among users of location based services, five times more use Instagram than Twitter [7]. To generalize from Twitter, we present a system that has low coupling between social medium post formats and the event detection system.

Our main contributions in this paper can be summarized as follows: 1. A novel framework utilizing hashtags and locations from Instagram and Twitter posts to detect and identify local ongoing events 2. Experiments on a real-world dataset which show the effectiveness of our event detection method.

2. APPROACH OVERVIEW

We shall consider social media posts consisting of a text, a timestamp, and an optional geographical coordinate. Both Twitter and Instagram posts comply with this specification; notice that any media such as images and videos are discarded. The overall approach is illustrated in Figure 1. Our method assumes that the language of all posts is English, and we attempt to remove all posts written in other languages using a naive Bayes classifier provided by Shuyo [5]. We also remove near-duplicate posts, which may be spam or posts cross-posted on several social media, as such posts do not provide more evidence on events happening. To efficiently identify near-duplicates, we use a locality sensitive hashing approach described by Leskovec et al. [4]. We extract location names based on a collection of place names from Wikipedia[1]. Posts are collected in a sliding window which contains post from within the past τ seconds. When the sliding window is changed sufficiently a snapshot of the posts in the window is sent to the second stage which is the actual event identification.

3. EVENT IDENTIFICATION

In order to detect local events, we exploit the properties that events are associated with a specific geographical location and that special hashtags are usually used to identify events. First, we extract hashtags and group posts sharing hashtags together. To keep the clusters specific, we ignore hashtags with an inverse document frequency lower than *minIdf*. Notice that one post may end up in multiple clusters if it has several hashtags. Since various hashtags

[1] https://dumps.wikimedia.org/enwiki/20151102/

WebSci '16 May 22-25, 2016, Hannover, Germany

ACM ISBN 978-1-4503-4208-7/16/05.
DOI: http://dx.doi.org/10.1145/2908131.2908197

may be associated with a single event, we afterwards combine overlapping hashtag clusters. Clusters with more than *minClusterSize* posts and a Jaccard similarity greater than or equal to *minOverlap* are combined until no two clusters have a similarity greater than *minOverlap*.

To find the location of events within a hashtag cluster we identify geographical activity hot spots. We apply the DBSCAN [2] clustering algorithm on all Geo-tagged posts in a hashtag cluster. The DBSCAN algorithm is used for two reasons: 1. The distance measure is user defined, which allows us to use geographical distance and hence makes it easier to reason about its parameters 2. Outliers, which we assume to be common in social media data, are automatically discarded. Since DBSCAN is used two times in this system, we denote the two parameters for this specific clustering as $minPts_1$ and eps_1. As mentioned, we use the geographical distance between two posts as the distance measure. While posts without a location cannot provide evidence on the location of an event, they may still provide useful insights in them. We therefore add all posts that do not contain a location to each of the resulting clusters of DBSCAN. We call these clusters candidates.

After location clustering, there are cases of multiple candidates for the same event within a small geographical distance. These occurrences are likely due to difficulties in picking a good similarity threshold (*minOverlap*) for hashtag clustering. We tackle this problem heuristically by performing a second clustering on the candidates from the previous step. Again we use DBSCAN with the geographical distance measure. However, we take the center coordinates of each candidate as input and do not consider any points to be noise (i.e. $minPts = 1$). As a result, the DBSCAN algorithm will group candidates which are located nearby within a distance of *eps*. Within each of these groups, we consider whether to merge candidates based on their overlap of hashtags, measured by Jaccard similarity, but use a lower threshold (*minHashOverlap*) this time. We denote the *eps* parameter for this DBSCAN clustering eps_2.

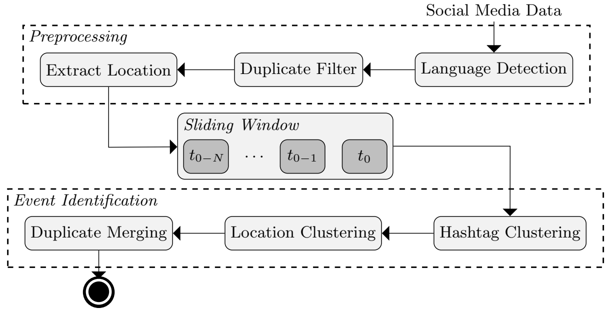

Figure 1: Overview of the event detection pipeline.

4. EXPERIMENTS AND RESULTS

As the system is split into different steps, experiments are conducted on the overall system in order to measure the performance. Precision and recall scores are used for the evaluation.

The primary dataset is collected in London on November 17 2015 from 20.00 to 21.00 UTC. We partition the post into three datasets: DS1 which consists of all the post gathered from Twitter, DS2 which consist of all the post gathered from Instagram, and DS3 which consists of all the posts from DS1 and DS2 combined. Each post has been manually labeled as an event or a non-event. The characteristics of the datasets can be seen in Table 1. We distinguish between large and small events. In addition, there has to be five posts about an event for our approach to be able to detect it.

	DS1	DS2	DS3
Source	Twitter	Instagram	Twitter & Instagram
#Posts	7096	2035	9131
#Posts w. Location	743	2035	2778
#Large Events	4	6	10
#Small Events	43	39	43

Table 1: Characteristics of the datasets.

	DS1	DS2	DS3
Precision	0.1667	0.2381	0.2000
Recall	0.5000	0.8333	0.6000
F_1	0.2500	0.3704	0.3000

Table 2: Result of event detection for large events.

Our experiments have been performed with the following parameter values: • $\tau = 3600$ s • $minIdf = 8$ • $minOverlap = 0.2$ • $minClusterSize = 5$ • $minPts_1 = 3$ • $eps_1 = eps_2 = 1000$ m • $minHashOverlap = 0.2$.

We test the event detection approach on datasets DS1, DS2, and DS3 to see which source of posts works best with our approach. We only consider large events when evaluating the clustering. We do this to evaluate the approach in relation to the results it could possibly achieve. The results for large events can be seen in Table 2.

Our approach performs best on DS2, i.e. using posts from Instagram only. Our approach identifies five out of six events in the dataset but also suggests 16 false positives. On DS1, only one out of four events is detected and ten false positives are suggested. When combining data from both sources (DS3), six out of ten events are detected.

In general, we find that most of the events found during the clustering are ongoing. In the DS1 dataset, all correctly detected events (two) are ongoing, and in the DS2 dataset, five out of six detected events are ongoing. In the DS3 dataset, 6 out of 7 correctly detected events are ongoing. Overall, it seems that a relatively short length of the sliding window ($\tau = 3600$ seconds) mostly avoids detection of passed events due to their low activity level. The estimated locations of correctly detected events are generally close to the true locations. All but one event are estimated to be located exactly at the true location. One event is placed 50 meters from its true location in DS1 and 400 meters in DS3. This event is largely based on Twitter posts, which indicates that the location data from Instagram is more reliable than that of Twitter.

5. CONCLUSION

In this paper we presented a novel method for local event detection using Twitter and Instagram. While the approach does not outperform contemporary approaches in terms of accuracy, the detected events are ongoing and their locations are estimated with a high precision. Boettcher and Lee [1] have achieved a high accuracy partly by using a classifier to remove false positives, so future work will include experiments using this classification approach.

6. REFERENCES

[1] A. Boettcher and D. Lee. Eventradar: A real-time local event detection scheme using twitter stream. In *GreenCom 2012*, Nov 2012.

[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. AAAI Press, 1996.

[3] R. Lee and K. Sumiya. Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection. In *SIGSPATIAL*. ACM, 2010.

[4] J. Leskovec, A. Rajaraman, and J. D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2 edition, 2014.

[5] N. Shuyo. Language detection library for java. https://github.com/shuyo/language-detection. Accessed November 18 2015.

[6] M. Walther and M. Kaisser. Geo-spatial event detection in the twitter stream. In *ECIR*, 2013.

[7] K. Zickuhr. Location-based services. http://www.pewinternet.org/2013/09/12/location-based-services, September 2013. Accessed September 18 2015.