Identifying Local Events by Using Microblogs as Social Sensors

M-Dyaa Albakour
University of Glasgow, UK
dyaa.albakour@glasgow.ac.uk

Craig Macdonald
University of Glasgow, UK
craig.macdonald@glasgow.ac.uk

Iadh Ounis
University of Glasgow, UK
iadh.ounis@glasgow.ac.uk

ABSTRACT

Local search is increasingly attracting more demand, whereby the users are interested to find out about places or events in their local vicinity. In this paper, we propose to use the Twitter microblogging platform to detect and rank local events of interest in real-time. We present a novel event retrieval framework, where both the contents of the tweets and the volume of the microblogging activity are exploited to locate an event happening in a certain area within a city that matches the user's interests as expressed in the form of a query. In particular, the framework measures unusual microblogging activities in a certain area and uses that as an indication of the occurrence of an event which is then used by the ranking function. Since the proposed event retrieval task is a new Information Retrieval (IR) task, we devise a methodology that is inspired by the conceptually similar IR problem of video segmentation to thoroughly evaluate our approach. Our evaluation is conducted on a set of tweets collected over a period of twelve days from different areas of London, as well as two sets of local events collected within the same period using crowdsourcing and local news sources in London. In addition to new insights on the factors that influence the development of an effective event ranking model, our empirical results show the promise and effectiveness of our proposed approach in identifying and ranking local events in real-time.

1. INTRODUCTION

It has been suggested that a large proportion of queries submitted to web search engines has a "local intent" and that these queries compose the majority of searches submitted from mobile phones [29]. Examples of information needs expressed by such queries include "what is happening near me?" or "finding restaurants in the Covent Garden district". This highlights the importance of building effective local search tools that serve this type of information need. In this paper, we investigate whether social media can answer local information needs where people are interested in finding about events of interest taking place in their local vicinity. Indeed, the communities of users in Twitter often share messages about local events as they progress [32]. To give the reader a concrete example of how local events are reflected in social media, we plot in Figure 1 the volume of tweets that are posted within London and contain the phrase "beach boys" over a period of 12 days, where "beach boys" is the name of a rock band who held a concert in London's Royal Albert Hall during the considered time period. We observe that just before and during the concert, tweets mentioning the "beach boys" within London have spiked. This is an indication that the concert as a real world event has been reflected in the tweeting activities within the city.

Recently, there have been some attempts to harness social media for event-based information retrieval (IR). This includes (i) identifying social media content relevant to *known* events [7, 26] and (ii) detecting *unknown* events using user-generated content in social media [8, 21, 27]. In the first case, social media content is identified to provide users with more information about a planned event (e.g. a festival or a football match). Users would be able for example to access tweets about ticket prices before the event, or Flickr photos posted by attendees after the event. The second case is more challenging as there is no prior knowledge about the events. While some approaches have focused on detecting news-related events [27], or simply clustering social media content based on a database of targeted events [8], a recent work has devised methods for retrieving global events from Twitter archives that correspond to an arbitrary query (event type); a problem which the authors called "structured event retrieval" over Twitter [21].

Unlike [21], which focused on non-local events, we make use of the opportunities that social media can bring to *local* search services. In particular, we define a new *localised* IR task that extends the aforementioned structured event retrieval task introduced in [21]. The task we propose aims at identifying and ranking local events based on social media activities in the area where the events occur. In other words, we use social media as a *social sensor* to detect local events in real-time.[1] In particular, the task involves answering a user query by ranking local events as inferred from social media. The query represents the information needs of the user looking for events of interest where they live or where they are at the moment. Each event retrieved in the ranked list is characterised by the area where it happened and its starting time. We treat this as a ranking task, as we do not only identify local events from social media but we also aim

OAIR '13, May 22-24, 2013, Lisbon, Portugal

[1]Treating social media as a social sensor has been suggested in previous work, for example [26] and the EU FP7 social sensors project `http://www.socialsensor.eu`

to find those that are relevant to the information needs of the user. A *relevant* event in this case can be an event that matches the user's interests as specified explicitly by the user's query. It can also be an event that matches the user's context as implicitly identified by the time of the query, the location of the user and/or her profile. In this case, the recency of an event or its distance to the user's location may decide its relevance. This task is novel and different from the previously mentioned attempts for identifying *unknown* events from social media in several ways. First, we consider only events which are of interest to the user (not just types or categories of events). Secondly, in addition to identifying the time of the events, we also estimate their location (i.e. an area in the city where they occur). It should be noted however that the location is a granular notion and the granularity can vary from buildings and streets to entire cities. Examples where this can be useful include helping tourists to find things to do whilst visiting a large city. In such situations, the tourist may issue queries to find music or sporting events and the system will identify and locate where music concerts or sporting contests are taking place and what people are saying about them in social media. In fact, this is the vision of future smart cities that provide their citizens with the capability to search for real world events happening around the city [2].

Moreover, we take the first steps to build the required infrastructure to perform this task. Although it can make use of any social (media) data, in this paper, we focus on the publicly available Twitter microblogging platform and present a novel framework for local event retrieval using Twitter. Our event retrieval framework works by representing each location (e.g. an area of the city) as a time series of tweets. It ranks pairs of *time points* and *locations* according to how well the query matches an event that may have occurred at the given point of time within a given location. This is achieved by combing evidence from (i) the textual content of the tweets and (ii) an analysis of change in the volume of tweets observed over time. We thoroughly evaluate our event retrieval framework using geo-located tweets from four different boroughs in London as well as from the entire metropolitan area of London over a period of twelve days. London is chosen because it is a big vibrant city, and according to Twitter Grader,[2] it is the top city in the world in terms of the Twitter user population. We use two different sets of queries, one collected using crowdsourcing and another one collected from local news feeds in each of the given four boroughs. The results are promising, showing the effectiveness of the framework in using microblogs as social sensors to correctly identify the time and the location of events.

The main contributions of this paper can be summarised as follows. First, we introduce the new IR task of local event retrieval. Then, we propose our novel event retrieval framework that is capable of using Twitter to produce a ranked list of local events as a response to a user query. We also devise an evaluation methodology for the local event retrieval problem by mapping it into the conceptually similar IR problem of video segmentation. Finally, our empirical results identify the factors that influence the effectiveness of local event retrieval. The rest of the paper is organised as follows. In Section 2, we present related work. Section 3 gives a formal description of the local event retrieval problem. In Section 4, we present our event retrieval framework, while Section 5 describes the evaluation process and the measures used. Section 6 describes the experiments and reports the results. Finally, Section 7 summarises our conclusions.

[2] http://tweet.grader.com/top/cities

Figure 1: A plot of the volume of tweets in London that contain the phrase "beach boys" over time.

2. BACKGROUND

Early work on event detection studied events in news streams. In particular, Topic Detection and Tracking (TDT) was introduced in the late nineties as an initiative for addressing the challenges in event-based organisation of broadcast news [3]. Topic detection or topic clustering is a major task studied in TDT, which aims to place a new story within the most appropriate news cluster that discusses the same event [4]. First story detection or new event detection is another problem tackled within TDT, where the aim is to identify the first story (article) in the news that discusses a new emerging event [31].

The emergence of social media has led to a huge growth in user-generated content, which triggered research on turning this content into useful knowledge. A reasonably representative list of such applications includes extracting high quality information from social media [1] and using social media as a prediction power [30]. In addition, social media was investigated for event identification e.g. [8, 27] and the aforementioned TDT tasks were revisited, e.g. first story detection in Twitter [23]. In the remainder of the section, we summarise previous research that uses social media to support event-based information retrieval and seeking. The main motivation behind these approaches is that people reflect in social media, e.g. Twitter, on news and what they are currently doing [14, 16]. However, the challenge is that a large percentage of tweets are about conversations and daily routine [14]. The latter issue does not exist in news streams where all incoming documents are actual stories. Moreover, not all tweets are credible, as a large percentage can originate from spammers and people retweeting rumours [9].

Several researchers worked on developing techniques for retrieving social media content related to *known* events, i.e. the actual event is known a priori. Becker *et al.* [7] presented approaches for generating queries issued to multiple social media sites to retrieve user-contributed contents associated to a known event already advertised on social platforms such as Facebook and LinkedIn. A more challenging problem to finding content about known events, is the use of social media as a sensor to detect *unknown* events, which is the most related task to our work. There have been few attempts to tackle this challenge. Sankaranarayanan *et al.* [27] introduced TwitterStand, a system that detects breaking news from Twitter online. They employ clustering with a textual classifier on tweets originating from news seeders (handpicked news agencies) to detect news-related tweets. Similarly, an online clustering framework was introduced in [8] for identifying unknown events in Flickr images where multi-

feature similarity metrics were employed. Their work is however similar to the TDT topic clustering task and limited to clustering social media content based on a database of events. Twitter Monitor is an online monitoring tool that detects *bursty* keywords in tweets by simply observing their frequency over time [19]. These bursty keywords are then used to rank daily trends in Twitter. In our proposed approach, we also use burstiness (the sudden change) as an evidence of an occurrence of events. Recently Metzler *et al.* devised a new IR task of structured event retrieval over Twitter in a non-localised manner [21]. They have also developed temporal query expansion models to rank segments of time corresponding to a query (a type of events) using the global tweets observed on those segments. The top ranked segments correspond to the detected events. Their approach also relies on the burstiness of a term to aid query expansion and to extract tweets that summarise a detected event. The task we define is an extension to [21], where in addition to identifying events, we also try to *locate* them and rank them as a response to a user query. Moreover, we devise a more suitable evaluation procedure for this task.

We are advancing the state-of-the-art for detecting and locating unknown events in social media and proposing a new IR task of local event retrieval. In the next section, we formally describe the problem of local event retrieval.

3. PROBLEM FORMULATION

The overall goal of this paper is to identify and rank local events happening in the real world as a response to a user query. For a formal definition of a *local event*, we adopt a definition that has been previously used in the TDT new event detection task over broadcast news [3]. This definition states that an event is something that occurs in a certain place at a certain time. Formally, we consider a set of locations $\mathcal{L} = \{l_1 , l_2, ...\}$ that are of interest to the user. The granularity of locations can vary from buildings and streets to entire cities. For example, we might consider each location to represent an area in a city in which the user is located. The city in this case is considered to be divided into equally sized areas specified by polygons of geographical coordinates, or we can use the divisions defined by the local authority such as postcodes or boroughs. Each location l_i at a certain time t_j is denoted by the tuple $\langle l_i, t_j \rangle$. We define the problem of local event retrieval as follows. For a user interested in local events within locations $\mathcal{L}$ (explicitly defined or implicitly inferred from the current user's location), the event retrieval framework aims to score tuples $\langle l_i, t_j \rangle$ according to how likely t_j represents a starting time of an event within the location l_i that matches the user query. An event is considered relevant if it matches the explicit query of the user and/or the implicit context of the user (the time of the query, the location of the user and or her profile). In other words, the event retrieval framework defines a ranking function that gives a score $R(q, \langle l_i, t_j \rangle)$ for each tuple $\langle l_i, t_j \rangle$ with regards to the user's query q.

In this paper, we use Twitter for local event retrieval, targeting local events happening in a city. Examples include festivals, football matches or security incidents. When expressed explicitly by a user, a *query* is assumed to be in the form of a bag of words (e.g. "live music", "conference"). A location l_i at a certain time t_j is characterised by the microblogging activities observed at that location within a given time frame $(t_j - t_{j-1})$. The microblogging activities are represented with a set of tweets originating from that location shared publicly within the given time frame $(t_j - t_{j-1})$. This set of tweets is denoted by $T_{i,j}$. Note that the fixed time frame is defined using an arbitrary sampling rate $\theta; \forall j : t_j - t_{j-1} = \theta$. An event happening in the real world is represented by a tuple $\langle l, t_s, t_f \rangle$; where l is the location where the event is taking place, t_s is the starting time and t_f is the finishing time. Our aim is to use the microblogging activities as the main source of evidence to define the ranking function $R(q, \langle l_i, t_j \rangle)$. More specifically and to define the ranking function, we use the set of tweets $T_{i,j}$, and a time series of tweets $\mathcal{T}_{i,j} = \langle .., T_{i,j-2}, T_{i,j-1}, T_{i,j} \rangle$ in the location l_i before the current time t_j. This allows us to identify sudden changes in the microblogging activities, which may have been triggered by an occurrence of an event. Moreover, the event retrieval framework can identify a subset of the tweet set $T_{i,j}$ that matches the query, which may help the user in the event information seeking process.

4. EVENT RETRIEVAL

In this section, we describe our event retrieval framework. The framework aims to define an effective ranking function that scores tuples of time and location according to how likely they represent the starting time and the location of a relevant event for a given query. Note that with regards to the previous definition of the local event retrieval problem in Section 3, as a first step, we are not aiming to determine the finishing time of an event. We recognise that for some applications the finishing time of an event may be important, e.g. surveillance applications, however we leave this for future work. As discussed in Section 3, we aim to use tweets as the main source of evidence to score the tuples. In particular, we define two components built on this evidence:

1. The first component is based on the intuition that social media may reflect real world events, hence when an event occurs somewhere we expect to find topically related social posts about it originating from the location where it occurs. To instantiate this component, for each location at a given time, i.e. for each tuple $\langle l_i, t_j \rangle$, we measure how much the tweets $T_{i,j}$ corresponding to the tuple are topically related to the query q.
2. The second component is based on the intuition that events trigger an increasing microblogging activity [32] causing peaks of tweeting rates during the event (bursts). For this component, we aim to quantify the change in the tweeting rate, the volume of tweets over time, observed at $\langle l_i, t_j \rangle$ when compared to previous observations over time at the same location. In other words, we aim to measure the unusual microblogging behaviour that may indicate an occurrence of an event. To compute the tweeting rate, we can either consider all the tweets posted within the given time frame at the given location or only a subset of those which are relevant to the user query, e.g. tweets which contain terms of the query.

Following this, the ranking function can be defined as a linear combination of the previous two components as follows:

$$R(q, \langle l_i, t_j \rangle) \propto (1 - \lambda) \cdot S(q, T_{i,j}) + \lambda \cdot E(q, \langle l_i, t_j \rangle) \quad (1)$$

where $S(q, T_{i,j})$ is the score of the tweet set $T_{i,j}$ that quantifies how much they are topically related to the query q; $E(q, \langle l_i, t_j \rangle)$ is a score proportionate to the change in the tweeting rate with regards to the query q at the given time t_j within the location l_i, and $0 \leq \lambda \leq 1$ is a parameter to control the contribution for each component in the linear com-

bination in Equation (1). Next, we show how we approach the problem of quantifying each component.

4.1 Aggregating Tweets

To estimate $S(q, T_{i,j})$ in Equation (1), we propose to borrow ideas and techniques originally designed for the IR problem of expert search. In expert search, a profile of an expert candidate is typically represented by the documents associated to the candidate [6, 18]. Similarly, the tuple $\langle l_i, t_j \rangle$ is associated with a set of tweets. Inspired by [18], the score of each tuple (candidate) can be estimated by aggregating the retrieval scores (votes) for each tweet (document) associated to it. In [18], several voting techniques were used to aggregate the scores. We use the intuitive, yet effective, CombSUM voting technique, which estimates the final score of the tweet set representing a tuple (candidate) as follows:

$$S(q, T_{i,j}) = \sum_{t \in Rel(q) \cap T_{i,j}} (Score(q, t)) \quad (2)$$

where $Rel(q)$ is the subset of tweets that match the query q and $Score(q, t)$ is the individual retrieval score obtained by a traditional bag-of-words ranking function, e.g. BM25 [24]. Higher scores represent more topically related tweets for the considered tuple.

4.2 Change Point Analysis

The problem of quantifying the score $E(q, \langle l_i, t_j \rangle)$ in Equation (1) maps well to *change point analysis*, a previously studied problem in the statistics literature, e.g. [13, 15]. Change point analysis aims at identifying points in time series data where the statistical properties change. It has been previously applied to detect events in continuous streams of data. For example, Guralnik *et al.* developed change point detection techniques that can accurately detect events in traffic sensor data [12]. In our case, the change point analysis can be applied on the tweeting rate in a location l_i to quantify the probability that the tweeting rate at a certain time t_j represents a change point when compared retrospectively to previous points in time $t_{j-1}, t_{j-2}, .., t_{j-k}$. We apply the Grubb's test [11] as a change point detection technique as it is computationally inexpensive and it has been successfully applied in a similar context, namely first story detection from Twitter and Wikipedia [22]. We leave for future work the investigation of other change point analysis techniques such as the ones described in [15]. Given a location l_i and at each point of time, e.g. on minute intervals, we maintain a moving window of size k points, e.g. k minutes, over the previous observations. We apply the Grubb's test to each moving window to determine if the tweeting rate of the last point is an outlier that stands out with respect to the tweeting rates of previous observations. With Grubb's test, r_j is an outlier if $v = (r_j - \overline{x}_{j,k})/\sigma^2 > z$, where $\overline{x}_{j,k}$ is the mean tweeting rate in the window $(t_{j-k},\ t_j)$, σ is the standard deviation of the tweeting rates in the window $(t_{j-k},\ t_j)$, and z is a fixed threshold. Note that this test gives a binary decision for each point in time. We smooth this binary decision into a normalised score and use it for the second component of Equation (1) as follows:

$$E(q, \langle l_i, t_j \rangle) = E_c(t_j) = 1 - e^{(\frac{-\ln 2}{z} \cdot v)} \quad (3)$$

where $0 \leq E_c(t_j) \leq 1$ represents a score of a change point using the Grubb's test. Note that when $v = z$, the resulting score in Equation (3) is equal to 0.5. As previously discussed, the tweeting rate r_j can be estimated in two different ways: (i) By simply using the volume of tweets posted in the given location within the time frame corresponding to t_j, i.e. $r_j = |T_{i,j}|$. We call this a *query independent* (QI) tweeting rate; and (ii) By using the score of the voting technique described in Section 4.1, i.e. $r_j = S(q, T_{i,j})$. We call this a *query dependent* (QD) tweeting rate.

It should be noted that our framework can operate in a *real-time* fashion where social feeds are incrementally indexed such that the retrieval components are able to provide the freshest results.

5. EVALUATION

The goal of our event retrieval framework is to rank tuples of time and location according to how likely they represent a starting time of an event that is both *correctly identified*, and *relevant* to the user's query. A correctly identified event is an event that has occurred in the given location and time frame corresponding to the tuple retrieved. Defining the *relevance* of an event can be done in several ways and is also dependent on the application. We identify two major factors that can determine the relevance of an event: (i) Matching the interests of the user as specified *explicitly* by the query. For example, a concert is relevant to the query 'music' but a football match is not; and (ii) Matching the user's context as *implicitly* identified by the location of the user, the time of the query and/or the user's profile. For example, an event which occurred two days ago or hundreds of miles away may be considered irrelevant if the application is an event search tool for tourists. In our evaluation in this paper, we will only consider the first factor to determine the relevance of a correctly identified event and we leave the other factor for future work.

Following this, to evaluate the effectiveness of our event retrieval framework, we need to assess the two different aspects of the framework, namely the capability to correctly *identify* events and the capability to *rank* highly tuples (of time and location) that represent relevant events. For the first aspect, and to focus the evaluation, we consider only a single location and test whether the event retrieval framework can correctly identify all the relevant events for a query that have occurred during a period of time in that location. We discuss the evaluation measures and procedures used for this aspect in Section 5.1. For the second aspect, we consider multiple locations and test the effectiveness of ranking tuples that represent relevant and correctly identified events. Similarly, we cover this aspect in Section 5.2.

5.1 Event Identification

To assess the event identification aspect of our framework, we measure how effective the framework is in correctly identifying all the relevant events to a query within a single location. In that case, the tuples have only the time dimension and the task is to identify the correct starting time for all relevant events within that single location. From an evaluation perspective, we consider this as a similar task to the shot boundary detection task in the Video Track, which was first introduced in TREC 2001 [28]. In particular, detecting the starting time of an event would be similar to detecting a shot boundary of a "dissolve" type which spans over multiple frames of video. In this evaluation, a dissolve effect is considered to be correctly identified if it overlaps with 50% of the correct one. Similarly in our case, and since we do not identify the finishing time of an event, we consider an event to be correctly identified if the detected starting time falls

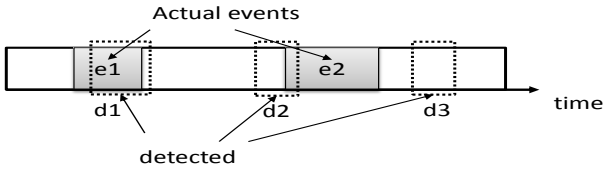

Figure 2: An illustration of the evaluation procedure for identifying all events within a single location. Shaded rectangles represent actual events, whereas dotted rectangles represent events identified by an evaluated system. In this case, N_A=2 (e_1, e_2), N_D=1 (e_2), N_I=2 (d_2, d_3) and N_C=1 (e_1).

within the first half of the actual event. Therefore, and following this assumption, we can reuse some of the measures introduced in [25, 28] for video-to-shots segmentation, which are discussed below, to evaluate our retrieval framework for correctly identifying events within one location. It should be noted that this evaluation works only if there is a single relevant event that happens in a given location at a particular time, which may not always be the case if the location has a coarse granularity (an entire city), e.g. several music concerts may be occurring within the same city. However, we are taking the first step towards evaluating the event identification aspect and we leave for future work an evaluation methodology and corresponding measures where we can consider multiple events happening at the same time within the same location. To estimate the measures used in the TREC Video Track [25, 28], we first quantify the following observations: N_A the actual number of events, N_D the number of events ignored (deleted) by the system, N_I the number of events inserted by the system and N_C the number of correctly identified events. The reader can refer to Figure 2 for a concrete example of what these quantifications are. Following this, we can estimate a number of measures for success and error [25]. Here, we identify the measures that can be suitable for the task and domain in hand, namely *error rate*, *recall* and *precision*.

The error rate can be estimated as in Equation (4):

$$ER = \frac{N_D + N_I}{N_A + N_I} = \frac{N_D + N_I}{N_C + N_D + N_I} \quad (4)$$

It gives a normalised quantification of the errors made by the system (not detecting actual events and inserting false events). For the example in Figure 2, the error rate is 0.75. However, this measure may not be adequate for the comparison of methods because it gives more importance to deleted events than to inserted ones (false detections) [25].

Recall and precision can be estimated with the aforementioned quantifications as follows:

$$R = \frac{N_C}{N_C + N_D}, \; P = \frac{N_C}{N_C + N_I} \quad (5)$$

For the example in Figure 2, the recall is 0.5 and the precision is 0.3. Note that to apply these measures, we need a certain cut-off point on the ranking list. The topmost tuples in the ranking list are the finite set of decisions made by the framework that will allow us to quantify N_D, N_I and N_C.

5.2 Event Ranking

In this evaluation, we aim to assess how accurate the retrieval framework is in ranking tuples that correspond to correctly identified events, which are also relevant to the user query. We can relax the evaluation to consider multiple locations instead of the single location constraint used in the previous case. As in Section 5.1, a retrieved tuple is considered to represent a correctly identified event if it falls within the first half of an actual event. Also, it has to correctly match the location of the event. Upon finding those tuples in the ranked list, we consider only those which are also relevant to the explicit user query in order to calculate a traditional IR ranking measure such as the precision at a certain rank k ($P@k$). In the next section, we will conduct experiments where we apply the evaluation procedures described for both *event identification* and *event ranking*.

6. EXPERIMENTS

The experiments examine the effectiveness of our framework in identifying and ranking events using Twitter. Specifically, we aim to answer the following research questions:

- RQ1: Can our proposed retrieval framework effectively use social activities on Twitter to detect events that match a user query and identify their starting time within a given geographical area of interest?
- RQ2: What is the contribution of the two components in our framework, as specified in Equation (1), on its performance to accurately detect and rank events?

In the remainder of this section, we first explain how we collected the datasets. Then, we describe the experimental setup and finally we report and discuss the results.

6.1 Datasets

There is no TREC-like dataset that would allow us to evaluate our event retrieval framework. Therefore, we have created two evaluation datasets. Each of our datasets contains geolocated tweets collected over time from different locations in the area of London. In addition, each dataset contains a set of queries and corresponding relevant events that occurred in those areas during the time in which the tweets were collected. In the following, we detail how the tweets and the events were collected to create both datasets.

Crawling Twitter: To create our two datasets, we crawled tweets using the Twitter streaming API.[3] The filter stream of the API enables us to crawl tweets posted from a certain area specified by a bounding box of geographic coordinates. It should be noted however that the majority of the tweets may not have geolocation information, Cheng *et al.* reported that only 20% of the tweets are coarsely geolocated (e.g. at a city level) [10]. Therefore, the tweets crawled only represent a portion of the tweets actually posted in a given region. In future work, we will consider applying tweet geotagging approaches, such as those described in [10], to obtain a larger subset of geolocated tweets for our task. Using the Twitter streaming API, two sets of tweets were crawled in a period of 12 days between Sep. 22^{nd} and Oct. 3^{rd} 2012. For the first set, we collected tweets from the entire metropolitan area of London, which resulted in 1,280,854 tweets. For the second set, we selected four boroughs in London: Croydon, Kingston, Richmond and Sutton. The crawling resulted in 842,552 tweets, each geolocated within one of the boroughs: 218,531 in Croydon, 178,893 in Kingston, 220,031 in Richmond and 225,097 in Sutton.

Collecting Events and Queries: To collect a set of queries and associated local events, we employ two different approaches. In the first one, we used crowdsourcing for identifying local events. In the second one, we selected local events reported in the RSS feeds of a local news agency.

[3] `https://dev.twitter.com/docs/streaming-apis`

Table 1: Events collected using crowdsourcing. The keywords describing the events are used as queries.

No.	event's keywords
1	iTunes Festival 2012
2	Liberal Democrat conference 2012 Clegg politics UK conference
3	MONDAY MUSIC NIGHT The Bedford MUSIC ROCK
4	Black History Month
5	Music Event orchestra of the age of enlightenment Southbank
6	Young believers choir concert
7	Kew Gardens death Woman killed by falling tree branch
8	The beach boys royal albert hall

With the crowdsourcing approach, we used Crowd Flower[4] for hiring workers to identify an event taking place in London. Using the Crowd Flower API, we created a task that asks workers to describe an event that is currently happening in London. In particular, the workers had to provide a title for an event they identify and a set of keywords that describe the event. They were also told to estimate the starting time of the event in a suitable time format. Following best practices for crowdsourcing [20], to increase quality control and avoid spam, we ask the workers to provide a URL that supports their answer, e.g. a web page that describes or mentions the event. The tasks were done during the same time of crawling the tweets and we collected 50 answers which we verified manually. 62% of the answers were spam and were therefore discarded. This high percentage of spam is due to the fact that we used text boxes that allow the workers to enter free text. We also discarded answers describing events which either happened in the past or were future events. We ended up with 8 valid answers. Each answer represents an event and a query corresponding to it. We considered the queries to be all the keywords that the workers used to describe the events. They are listed in Table 1. It should be noted that we could also use a subset of the keywords to represent a query. In future work, we aim to better understand what the queries in such a system typically look like.

The Guardian RSS feeds of local news[5] is used as another source of local events. The RSS feeds were collected for each of the four London boroughs (Croydon, Kingston, Richmond and Sutton) during the same period of crawling the tweets. We manually identified news articles in the RSS feeds of each borough describing local events that have occurred in the borough. Each local event has a location, which is the borough corresponding to the RSS feed from which the article was collected. We considered the title of the article to be a representative query. This has resulted in a total of 12 queries which are listed in Table 2. For both sets of events, we manually defined the date, the starting and the finishing time of each event when possible by examining the URLs provided by the workers or the content of the local news. For events where we could not specify the starting time, we considered the event as spanning over the entire day. Note that we do not aim to identify the finishing time of events. However, we still need both a starting time and a finishing time for each event to be able to conduct the evaluation described in Sections 5.1 and 5.2

Finally, we ended up with two datasets denoted by: (i) the (CS1) dataset comprising the first set of tweets from the entire metropolitan London and the crowdsourced events with their corresponding queries; and (ii) the (LG4) dataset comprising the second set of tweets from the 4 boroughs and the local Guardian events with their corresponding queries.

[4] http://crowdflower.com

[5] http://www.yourlocalguardian.co.uk/news/

Table 2: Events collected from RSS news feeds. The title of news articles are used as queries.

No.	Title of the news article
1	Olympic cyclist Joanna Rowsell launches St Raphael's midnight ladies walk in North Cheam
2	Hospital volunteers honoured at tea party
3	Proposed Worcester Park mosque causes friction in Lib Dem quarters
4	Richmond parents campaign against cost of travelling to Strode's College
5	Hampton and Twickenham mourns passing of former councillor and popular actress Lynne Ferguson
6	Richmond's Oscars rewards stunning performances
7	Kingston town centre gridlocked due to traffic chaos
8	Major delays on the A243 after rush hour collision outside Chessington Garden Centre
9	Catch the jazz fab four Simon Spillett John Critchinson Paul Morgan and Clark Tracey
10	Croydon North MP Malcolm Wicks dies after cancer battle
11	Ted Brown Black gay rights activist to talk in Croydon
12	Campaigners stage protest at Purley petrol station

6.2 Experimental Setup

To answer our research questions (RQ1 and RQ2), we conduct two experiments on the two datasets we have collected. In the first experiment, we consider only one location covering the entire metropolitan area of London. We run our retrieval framework using the queries and the tweets in the CS1 dataset. In this experiment, we aim to assess the performance of the retrieval framework for both *event identification* and *event ranking*. For event identification, we measure the performance using the measures specified in Section 5.1. For event ranking, we also follow the procedure in Section 5.2. However, as we have only a single relevant event for each query, we use the mean reciprocal rank (MRR) of the corresponding tuple in the ranked lists as an evaluation measure. In the second experiment, we consider four different locations which are the four London boroughs mentioned earlier. We run our retrieval framework using the queries and the tweets in the LG4 dataset. In this experiment, we aim to assess the performance of our retrieval framework for *event ranking* across multiple locations. We apply the evaluation procedure in Section 5.2. For the same reason noted above, we also use the MRR as an evaluation measure.

In both experiments, we sample the tweets with a sampling rate of $\theta = 15$ minutes. Note that the time frames can be defined on either a coarser scale (e.g. an hour) or a finer scale (e.g. a minute) depending on the application. For example, in the case of detecting emergency events we may want to consider 1-minute time frames. Following this and for the period of 12 days, each location is represented with 1152 points of time. The retrieval framework is instantiated as follows. To score the individual tweets for a query, we use the effective DFReeKLIM weighting model of the Divergence from Randomness framework designed particularly to rank short texts. It was one of the most effective tweet ranking approaches submitted to the TREC 2011 Microblog Track [5]. The parameters of the Grubb's test are set according to the experimental setup used in [22]. In particular, the threshold z is set to 3.5 and the window size is set to 10. We vary the parameter λ in Equation (1) between 0 and 1 increasing it by 0.1 at a time to assess the extent to which each component is important for the performance of the framework (see RQ2). Also, we experiment with the two different ways to measure the tweeting rate for estimating the change point score in Equation (3) which are the QD and the QI tweeting rates (see Section 4.2).

Table 3: Average scores obtained for two event identication measures at two cut-off points using the dataset CS1. When using the change point component, we report the best results for the different values of λ. Bold figures denote the best run.

	cut-off k=1		cut-off k=3	
Framework setup	ER	P	ER	P
$\lambda = 0$	0.875	0.125	0.917	0.083
QD (λ=0.7)	**0.500**	**0.500**	**0.833**	**0.167**
QI (λ=0.5)	0.875	0.125	0.917	0.083

6.3 Results and Discussion

To answer our first research question (RQ1), we consider the results obtained for the *event identification* measures as described in Section 5.1. To calculate these measures, and as discussed before, we need to consider a cut-off point on the ranking list of tuples. Given that, in our datasets, there is only one relevant event for each query, we consider the two cut-off points ($k = 1$ and $k = 3$). In Table 3, we report the measures at the considered cut-off points for the first experiment (CS1). Observing the values of the measures for the cut-off point $k = 1$, we see that overall the results are promising. In the second row, the framework is capable of achieving a satisfactory performance with regards to both the error rate and the precision. In fact, these results show that our event retrieval framework is capable of correctly detecting the correct time of the target event for 50% of the queries ($P = 0.5, ER = 0.5$). When observing the values of the measures for the cut-off point $k = 3$, we see that the performance degrades and this is mainly because of the nature of our dataset where we only have one target event to detect. Overall, for our first research question (RQ1), our empirical results show that, even with the current low ratio of geotagged tweets, we can use Twitter as a social sensor to identify real world events that match a user query.

The results in Table 3 also give some insights for answering the second research question (RQ2). We see that the best results for *event identification* are achieved when using the change point analysis on the QD tweeting rate over time. Moreover, the best results are actually obtained when $\lambda = 0.7$ in Equation (1). This suggests that the change point analysis component of the QD tweeting rate has an important role on the performance of our framework to correctly identify events. However, when using the QI tweeting rate, the results at best are identical to the ones when the change point component is not applied ($\lambda = 0$), i.e. it has no contribution to the final score in Equation (1).

Now we consider the second part of the research question (RQ2), which deals with the *event ranking* performance. We report the MRR for both experiments in Table 4. The results for the first experiment (CS1) are reported in the first column of Table 4. The MRR scores are relatively high, which means that the framework is capable of ranking events within one location. Moreover, the results clearly show the advantage of using the change point analysis on the tweeting rate observed over time to rank events. In particular, the best performance is achieved when the change point analysis is applied on the QD tweeting rate, which identifies points of time where the query score peaks with regards to its score at previous times. To understand the contribution of the change point analysis to the ranking function, we plot the MRR scores for the different values of λ in Figure 3. In this case, the best results are actually obtained when giving more importance to the change point score than to the retrieval score of the tweet set obtained with the voting model (when $\lambda = 0.7$). However, from the error bars, we can see a high variance of the achieved MRR across different queries. The reason could be that some of the events are not reflected properly in the microblogging activity and therefore they are not detected by the retrieval framework in the first place. From the last row in Table 4, we observe that using the QI tweeting rate is not as effective, although it marginally improves the results. In this case, when we examine the MRR scores for different values of λ, the changes are marginal and therefore they are not plotted as for the QD case. In the second column of Table 4, we report the results for the second experiment (LG4). We observe that the performance for the second experiment (LG4) degrades when compared to the previous one (CS1). Also, the changes in the performance across different values of λ are marginal when considering either of the tweeting rates. However, the task here is harder for a number of reasons. First, we consider a number of smaller areas (finer-grained locations) within a city. In addition, the queries are different as they were collected from local news articles, which means that the events probably did not attract as much Twitter coverage as those identified with crowdsourcing. In fact, to quantify the popularity of those events, we have examined the web pages of those news articles, which contain a Tweet Button.[6] None of those articles attracted a single click, which is an indication of the low coverage of such events in Twitter. On the other hand, events collected in CS1 with crowdsourcing (Table 1) have more online coverage because the workers have found them on the web and therefore they attracted more interaction on Twitter. This may be an advantage for an end user application to favour potentially popular events that attract more social media coverage.

Table 4: MRR scores for the different runs. Bold figures denote the best run.

Framework setup	CS1	LG4
$\lambda = 0$	0.2343	0.0544
QD (best λ)	**0.5306** (λ=0.7)	0.0581(λ=0.2)
QI (best λ)	0.2740 (λ=0.5)	**0.1091**(λ=0.1)

To summarise the results for RQ2, we conclude that both components specified in Equation (1) are important for the event identification and ranking performance. However, the performance degrades when considering multiple locations of interest at a finer-grained division of a city (LG4) and the empirical results, in this case, are not decisive on which component of the framework has a higher importance. This is mainly because of the sparsity of the tweets in smaller areas and the nature of the considered events.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed to use Twitter as a social sensor that can identify local events happening in the real world. We have devised a novel event retrieval framework that is capable of identifying and ranking local events in a response to a user query. The retrieval framework combines evidence from the content of the tweets and the change point analysis on the microblogging behaviour to accurately identify

[6]A Tweet Button is a widget that allows users to easily share a link on a website and count the number of times it has been used: see `https://dev.twitter.com/docs/tweet-button`

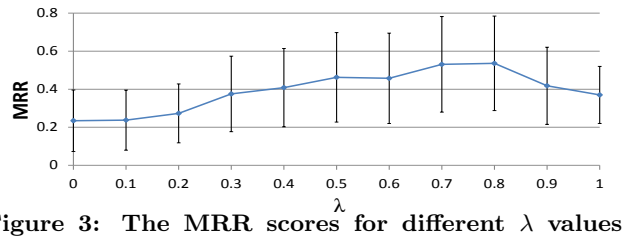

Figure 3: The MRR scores for different λ values when using the CS1 dataset and QD tweeting rate.

and rank local events. Moreover, we devised an evaluation methodology inspired from the conceptually similar IR problem of video shot segmentation. Our empirical results suggest that detecting local events using geo-located tweets is feasible but difficult. In particular, the results show that our event retrieval framework is capable of identifying and ranking events within a city. However, when applied on multiple fine-grained areas within the city, the retrieval effectiveness of the framework degrades, possibly because of the nature of the events considered in our experiments, i.e. their low coverage on Twitter.

For future work, we aim to extend the framework to deal with the caveats we observed in our evaluation. For example, when considering small areas in a city and to tackle the sparsity issue, tweets in nearby locations can also be exploited to improve the ranking function (e.g. by using the tweets in those locations for smoothing).

Acknowledgments

This work has been carried out in the scope of the EC cofunded project SMART (FP7-287583).

8. REFERENCES

[1] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media. In *WSDM'08*.

[2] M.D. Albakour, C. Macdonald, I. Ounis, A. Pnevmatikakis, and J. Soldatos. SMART: An open source framework for searching the physical world. In *OSIR at SIGIR'12*.

[3] J. Allan, J.G. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking. pilot study final report. In DARPA'98.

[4] J. Allan, S. Harding, D. Fisher, A. Bolivar, S. Guzman-Lara, and P. Amstutz. Taking topic detection from evaluation to practice. In *HICSS'05*.

[5] G. Amati, G. Amodeo, M. Bianchi, G. Marcone, C. Gaibisso, A. Celi, C. De Nicola, and M. Flammini. FUB, IASI-CNR, UNIVAQ at TREC 2011. In *TREC'11*.

[6] K. Balog, L. Azzopardi, and M. de Rijke. A language modeling framework for expert finding. *Information Processing & Management*, 45(1), 2009.

[7] H. Becker, D. Iter, M. Naaman, and L. Gravano. Identifying content for planned events across social media sites. In *WSDM'12*.

[8] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *WSDM'10*.

[9] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on Twitter. In *WWW'11*.

[10] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: a content-based approach to geo-locating Twitter users. In *CIKM'10*.

[11] F. Grubb. Procedures for detecting outlying observations in samples. *Technometrics*, 11, 1969.

[12] V. Guralnik and J. Srivastava. Event detection from time series data. In *SIGKDD'99*.

[13] L. Horváth. The maximum likelihood method for testing changes in the parameters of normal observations. *Annals of statistics*, 21(2), 1993.

[14] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *WebKDD'07*.

[15] R. Killick, P. Fearnhead, and I.A. Eckley. Optimal detection of changepoints with a linear computational cost. *arXiv*:1101.1438, 2011.

[16] B. Krishnamurthy, P. Gill, and M. Arlitt. A few chirps about Twitter. In *WOSN at SIGCOMM'08*.

[17] G. Luo, C. Tang, and P. S. Yu. Resource-adaptive real-time new event detection. In *SIGMOD'07*.

[18] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *CIKM'06*.

[19] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the Twitter stream. In *SIGMOD'10*.

[20] R. McCreadie, C. Macdonald, and I. Ounis. Identifying top news using crowdsourcing. *Information Retrieval*, 2012. Online.

[21] D. Metzler,C. Cai and E. H. Hovy. Structured event retrieval over microblog archives. In *HLT-NAACL'12*.

[22] M. Osborne, S. Petrovic, R. McCreadie, C. Macdonald, and I. Ounis. Bieber no more: First story detection using Twitter and Wikipedia. In *TAIA at SIGIR'12*.

[23] S. Petrović, M. Osborne, and V. Lavrenko. Streaming first story detection with application to Twitter. In *HLT/NAACL'10*.

[24] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Information Retrieval*, 3(4), 2009.

[25] R. Ruiloba, S. March, and G. Quenot. Towards a standard protocol for the evaluation of video-to-shots segmentation algorithms. In *CBMI'99*.

[26] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *WWW'10*.

[27] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. Twitterstand: news in tweets. In *GIS'09*.

[28] A. F. Smeaton, P. Over, and R. Taban. The TREC-2001 video track report. In *TREC'01*.

[29] G. Sterling. Study: 43 percent of total Google search queries are local. http://searchengineland.com/study-43-percent-of-total-google-search-queries-have-local-intent-135428. Accessed: 5 October 2012.

[30] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with Twitter: What 140 characters reveal about political sentiment. In *ICWSM'10*.

[31] Y. Yang, T. Pierce, and J. Carbonell. A study on retrospective and on-line event detection. In *SIGIR'98*.

[32] A. Zubiaga, D. Spina, V. Fresno, and R. Martínez. Classifying trending topics: a typology of conversation triggers on twitter. In *CIKM'11*.