3

Searching the Internet of Things

Richard McCreadie[1], Dyaa Albakour[2], Jarana Manotumruksa[1], Craig Macdonald[1] and Iadh Ounis[1]

[1]University of Glasgow, UK
[2]Signal Media, UK

3.1 Introduction

Despite the huge success of Web search engines, searching the Web is far from being a solved problem (e.g. see [64] by Yahoo! Search). However, the information needs of the searchers are increasingly 'time-sensitive' – about events happening now – and/or 'local' – where the user's location has some geographical bearing on the content that is relevant to their information need(s). For instance, events that are happening now or recently may have an impact upon the searching behaviour of users. Indeed, a search engine can detect a power cut in New York within seconds, based on the querying behaviour of mobile and nearby users [30]. However, while a Web search engine can retrieve many forms of online information, it can only sense real-world events through their impact on the online world (e.g. news stories, tweets, increased query volume).

In this chapter, we describe how real-world information needs can be better addressed by search engines through harnessing sensing infrastructures, including those from the Internet of Things (IoT). Indeed, the introduction of IoT sensors within the search engine provides more responsive/timely information than existing evidence sources, such as Web or social streams, as illustrated in Figure 3.1. For instance, by considering IoT sensor outputs such as real-time rain levels, a search engine can produce a more customised answer to queries such as "what is the current weather at JFK airport?". Furthermore, information needs such as "what is happenning near me?" (local event retrieval) can be better answered by fusing social media trend data (also known as social sensing) with physical sensor observations. For example, a

Figure 3.1 Data sources available to an IoT-connected search engine.

party in the town square might be detected through a combination of people posting about it on Twitter, with crowd density analysis over CCTV camera feeds from the square. It is also possible to provide recommendations (e.g. for tourists) about points-of-interest to visit, that are appropriate to particular personalised interests of the users and their current contextual situation (time, location, travelling with friends, etc.). In doing so, search engines can help users satisfy new forms of information needs needs centred on real-world events.

In the following sections, we first provide details of search infrastructure technologies suitable for obtaining and indexing observations from a plethora of diverse IoT-connected sensors (Section 3.2); Later, we show how physical sensor information and socially-sensed information – combined with such technologies – can be adapted for tasks such as local event retrieval (Section 3.3), event topic identification (Section 3.4) and venue recommendations (Section 3.5). We conclude this chapter by discussing the outlook for the field and some interesting future directions and applications for IoT technologies in the search domain (Section 3.6).

3.2 A Search Architecture for Social and Physical Sensors

To achieve effective and efficient search over sensor data streams, it is important to have a suitable search architecture. Early exploration into the sensor space focused on the development of tools and techniques for searching sensor data using classical information retrieval techniques and architectures [17, 28]. These approaches exploit sensor ontologies [46] in order to decouple user queries from the low-level details of the underlying sensors. For instance, they might map a rain gauging data stream to particular weather-related queries, such that current rain data can be displayed when a user enters one

of those queries. However, these ontologies are quite brittle in the face of changes in the user's query semantics and need to be hand-crafted for each domain/sensor stream. Hence, they cannot provide effective search over the arbitrary large and diverse sources of multimedia data derived from both physical and social sensors. Furthermore, sensor integration is only one of the components that are needed when building an IoT-enabled search engine. It is also critical to have effective and efficient indexing and retrieval processes over the sensor data, as well as have the ability to leverage the new search capabilities to build applications beyond the classic 'search bar'.

More recently, the SMART (Search engine for MultimediA enviRonment generated contenT) project[1] developed a framework [4] that aims to solve these issues, by providing an infrastructure where multimedia sensing devices in the social and physical world can be easily integrated into a central search system. By doing so, each sensor can provide information about their environment (physical or social) and make it available in real-time for search. As one of the most modern IoT search platforms in use today, we summarise the components that comprise SMART in Section 3.2.1. We then discuss some of the key challenges when building and deploying an effective IoT search engine like SMART in Section 3.2.2

3.2.1 Search engine for MultimediA enviRonment generated contenT (SMART)

SMART [4] is a framework designed to enable multimedia IoT sensing devices, both social and physical, to be integrated into a real-time search system. The architecture of the SMART framework is comprised of three distinct layers, as illustrated in Figure 3.2. At the lowest level we have the sensing devices that provide the physical world data. The edge node represents the software layer that processes the raw sensor data to produce metadata about the environment, which is streamed in real-time to the search engine using an appropriate representation. Examples of processing algorithms can include crowd data analysis for video streams or speech recognition in audio streams. The search layer collects the metadata streams from the various edge nodes and indexes them in real-time using an efficient distributed index structure. It also employs an event detection and ranking retrieval model that uses features extracted over the metadata streams to satisfy the user's information need. For instance, as will be described later in our discussion on search applications in

[1] http://www.smartfp7.eu/

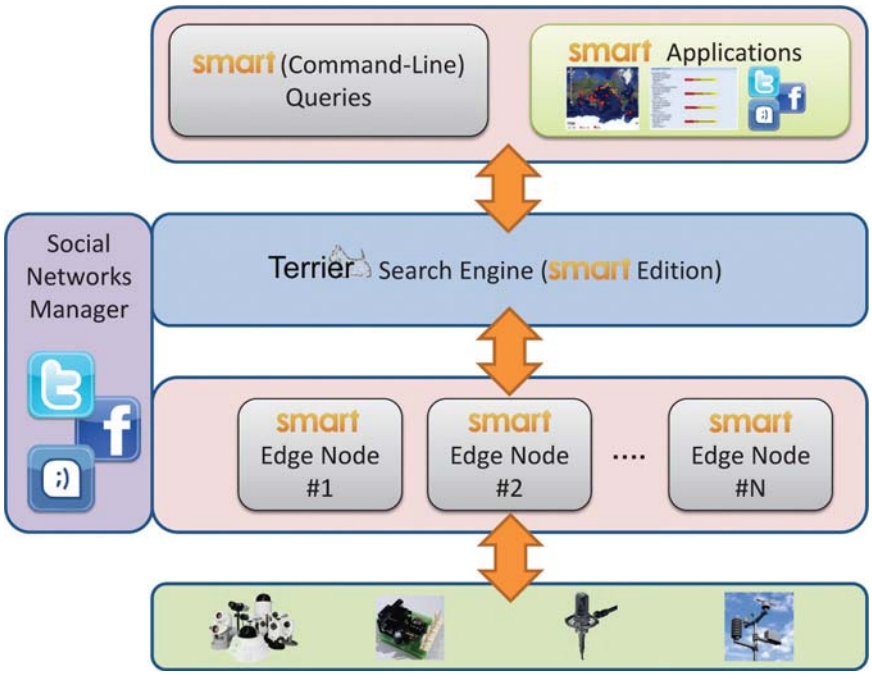

Figure 3.2 Architecture of the SMART framework.

Sections 3.3, and 3.4, the search service can be used to rank real-world events detected from the sensor streams that a user might be interested in attending. Queries can be either directly specified, or anticipated by the search layer using contextual information about the user, e.g. the user's location or their social profile. Finally the *application/visualisation layer* at the top offers reusable APIs to develop applications that can issue queries to the SMART engine and process or visualise the results. We further describe these three layers in more detail below.

Edge Node Layer

The edge node is the interface of SMART with the physical world. Each edge node can cover sensors from a single geographic area, e.g. a building block or a public square in the city centre. At each edge node, signal streams are processed to extract events/patterns that might be of value for answering one or more information needs. The signal streams can either be derived from physical sensors (e.g. audio/visual streams or environmental measurements), or from real-time Web crawling/social network streams. To achieve this, the

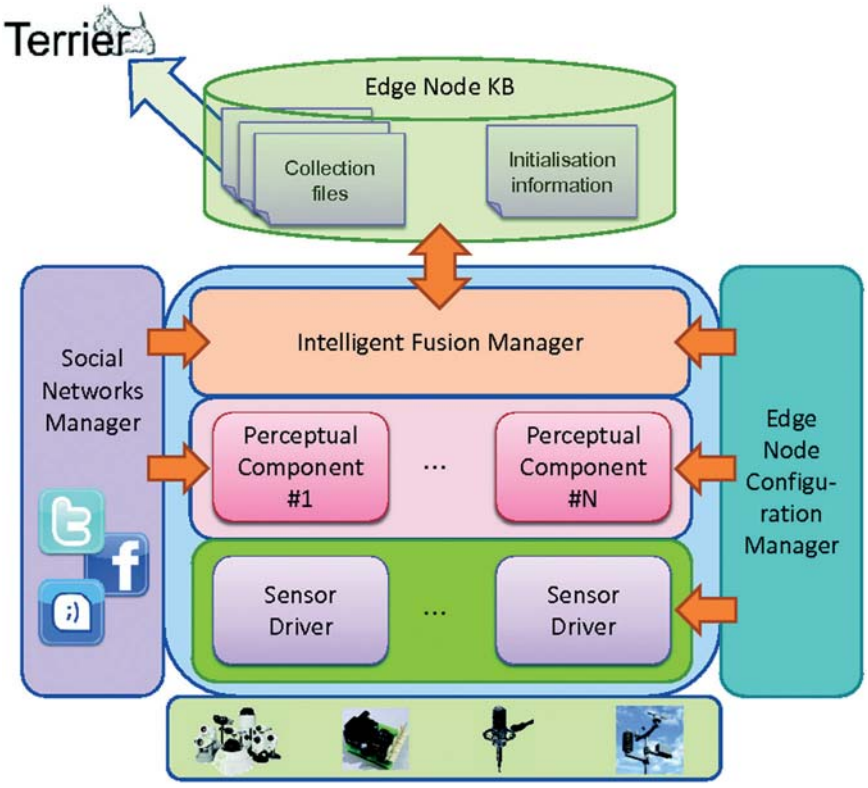

Figure 3.3 Edge node components.

design of the edge node is influenced by state-of-the-art IoT platforms and Linked Data techniques. The edge node architecture is shown in Figure 3.3. As we can see from Figure 3.3, at the lowest level of the edge node lies the sensors themselves. These sensors are interfaced with via sensor drivers, that allow for the connection to the sensor and the streaming ingestion of the raw sensor data into the edge node. The raw sensor data is then subject to processing by one or more perceptual components, which convert that data into a form that is more actionable. For instance, a perceptual component for an air quality sensor might take a CO2 reading and convert it to a label such as 'Normal' or 'High' based on external knowledge about what CO2 levels are acceptable. Next, the processed sensor outputs are sent to the Intelligent Fusion Manager of the edge node. This manager enables the reasoning over the outputs of different sensors within that edge node concurrently. For instance,

for an edge node responsible for tracking a shopping street, with physical CCTV camera streams spaced along that street and a social sensor looking for posts geo-located within that street, the manager might merge crowd signals from the CCTV cameras with the volume of social media activity to predict the number of people currently shopping there. Finally, the output of the intelligent fusion manager and the perceptual components feed an edge node knowledge base, which stores the observations made over the sensors across time. For instance, continuing the shopping street example above, the knowledge base would store the population estimates for the street at different times of the day. The edge node knowledge base content is stored as series of collection files that can be indexed by the search layer.

Search Engine Layer

The SMART search layer indexes in real-time streams of collection files from edge nodes, along with other conventional streams (such as social network posts or Web documents). It is built using the Terrier[2] open source search engine [48] with enhanced real-time indexing and a scalable distributed architecture to handle the large amount of streams. The SMART search layer is comprised of 7 core components as illustrated in Figure 3.4. The Indexing Component is responsible for the representation, storage and organisation of the information streams provided, such that they are available for later retrieval. It ingests the streams of collection files from edge nodes and social/Web documents (via data feed connectors), and performs a real-time indexing of those streams into appropriate data structures that allow for efficient retrieval. Real-time indexing ensures that as soon as an item (such as a social media post or street density summary) arrives on one of the input streams, that item will be searchable immediately. The index is distributed across multiple index shards (machines) so as to cope with a potentially high number and volume of social sensor streams, ensuring the scalability of the overall system architecture. This is achieved through the use of the distributed stream processing platform Storm.[3] Storm is one of the new generation of distributed real-time computation platforms, which provides an easy means to distribute complex software topologies across multiple machines, while maintaining fault tolerance and low management overheads. In this case, the content indexing pipeline is represented as a series of processing nodes (known as 'bolts'), where each node/bolt can be replicated and distributed across a local

[2]http://terrier.org

[3]http://storm.apache.org/

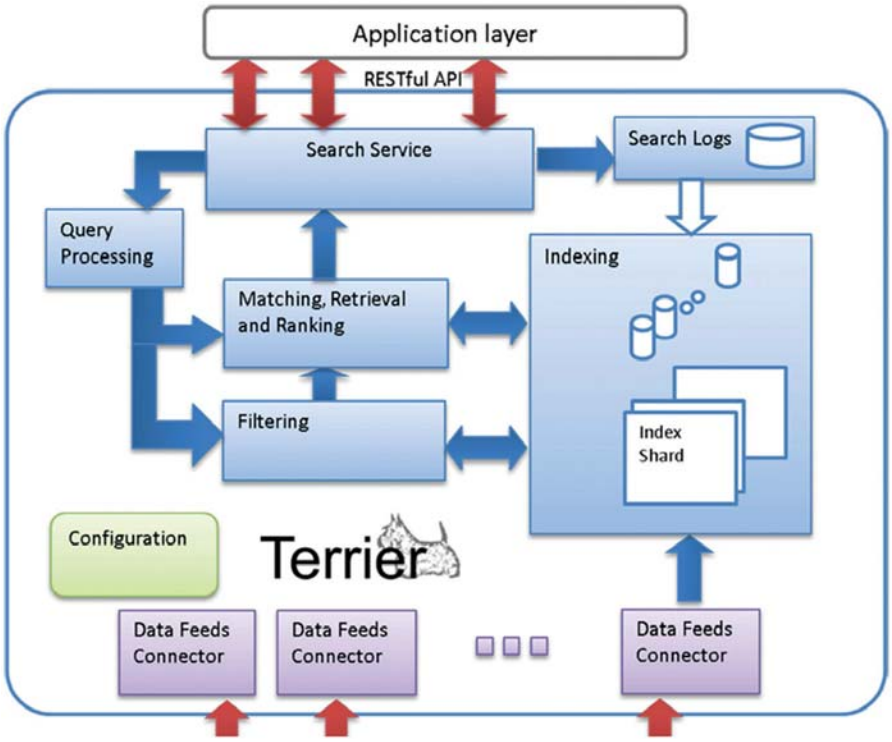

Figure 3.4 Components overview of the SMART search layer.

or cloud service machine cluster to achieve scalability. Next, the query processing component identifies the user information needs as specified explicitly by the user. Queries can be anticipated or expanded by observing past occurring patterns. The Matching, Retrieval and Ranking Component is responsible for matching explicit or implicit user queries against the index to *identify, rank* and *recommend* events/locations according to how they satisfy the users' information needs. This component relies on newly developed retrieval and recommendation models that can identify interesting "unusual" events across sensor (inc. social) metadata streams. The Filtering Component identifies in real-time events (or social network posts) as they happen that match a user's running query. This permits a user to be notified of new events that they will find interesting. This component handles queries after they have been submitted to the SMART search layer (as running queries) so that updates are streamed back to the higher level applications in real-time. The Search Logs Component maintains a recording of the search behaviour of

the user population. The search behaviour includes the user interactions with the application such as the queries that have been issued in a user search session, which search results have been displayed and any documents clicked by the user. This implicit feedback obtained by monitoring the users' search behaviour can be fed back into the SMART search layer, for example, to improve the effectiveness of the search results or the recommendations. The Search Engine API Component provides an interface to the SMART search layer where the main functionalities (search and running queries) are defined and are made available to higher level applications or services. Finally, the Configuration Component offers a series of administrative functionalities, such as the setup of the data streams to use as input and the choice of the matching algorithms to deploy.

Application Layer

The top layer of the SMART platform contains the software applications that can deliver the real benefits of the framework to the end-user. The application layer mainly supports developers who want to create Web 2.0 services or smart phone applications that exploit the framework capabilities. For example, the application layer includes open source end-user web applications that offer user interfaces to issue queries explicitly, or implicitly using the user context, to the search engine API and receive in real-time up-to-date results (events). In addition, it includes open source mashups that use the search layer visualisation APIs to display newly-breaking events, such as real-time balloon pop-ups on a map.

3.2.2 Challenges in Building an IoT Search Engine

Importantly, there are a variety of challenges when implementing an IoT search architecture like SMART. First, data stream collection and processing algorithms are needed to provide a uniform means to interface with a wide array of sensor types and to perform processing on those sensors' output to make that output interpretable/useful to the search engine. For instance, a raw video feed cannot be directly used to answer a user's information need. However, processing that feed through crowd analysis software to get crowd density for a street might be useful to predict the number of people visiting the area. Furthermore, some types of sensor streams require pre-filtering to make them useful. For example, it might be advantageous to define a social sensor, by filtering down a wider stream of posts to only those from a particular geographical region [2]. Within SMART, functionality like this is performed

by the perceptual components within each edge node. However, to incorporate the ever-growing range of IoT devices, new processing algorithms tailored to these devices will be needed.

Second, a common metadata model is needed to enable the processed sensor outputs to be mapped into a standardised metadata stream [12]. This is important, since often individual sensors only act as weak indicators of some higher level activity that the user might want to search for. For instance, if we want to detect live music in a city square, we might want to combine evidence from social sensors like discussions on Twitter, with physical evidence such as a locally captured audio or crowd density analysis from the square (c.f. Section 3.4). The use of a common metadata model can facilitate concurrent reasoning across multiple sensor streams by mapping lots of weak metadata signals from different sensors into the same format. For instance, SMART uses a model based on the OGC's Sensor Web Enablement standards [13] within the Intelligent Fusion Manager to achieve this.

Next, within the search engine itself, the efficient real-time indexing of the underlying metadata streams is critical. In particular, in an IoT environment, thousands of sensors can be feeding the search engine concurrently, and users expect the most up-to-date results. Hence, the search engine needs to be able to ingest high volume sensor streams in real-time while concurrently serving search requests over the most recent data. To achieve this, distributed stream processing platforms such as Storm[4] or Apache Spark[5] are used, as they allow for the low-latency processing of content in a distributed scale-out manner.

Finally, the types of queries and underlying information needs within the IoT search space are markedly different to those observed within a classical Web search domain. As a consequence, new retrieval models designed for these novel information needs are required. For instance, for an event search engine, a model that can effectively rank current (and possibly predict future) events based on criteria such as relevance, interestingness to the user or timeliness, are needed. Furthermore, in some applications, such as venue suggestions (that we will cover in detail later in Section 3.5), additional criteria needs to be considered, such as the user location (and hence distance to the event) and other contextual features such as the time of the day or the current weather. Current systems rely on state-of-the-art learning-to-rank techniques [39] to learn an effective combination of these diverse types of evidence when ranking.

[4] https://github.com/nathanmarz/storm/

[5] http://spark.apache.org/

In the remainder of this chapter we discuss three recent applications of the SMART framework that examined how to satisfy new user information needs using social and IoT sensing. In particular, we discuss social sensing with SMART for event retrieval in Section 3.3. Section 3.4 describes an application where IoT sensor streams were fused with social evidence for event topic identification. Finally, we discuss context sensitive venues-recommendation based on social sensing in Section 3.5.

3.3 Local Event Retrieval

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 [58]. Examples of information needs expressed by such queries include "what is happening near me?" or "finding restaurants in the Covent Garden district". The prevalence of such queries highlights the importance of building effective local search tools that serve this type of information need. In this section, we present an approach for local event retrieval, where we rely solely on social media as a *social sensor* to detect events in real-time.

3.3.1 Social Sensors for Local Event Retrieval

Our motivation stems from the fact that the communities of users in Twitter often share messages about local events as they progress [66]. To give the reader a concrete example of how local events are reflected in social media, we plot in Figure 3.5 the volume of tweets that are posted within London and contain the phrase "beach boys" over a period of 12 days, where "beach boys"

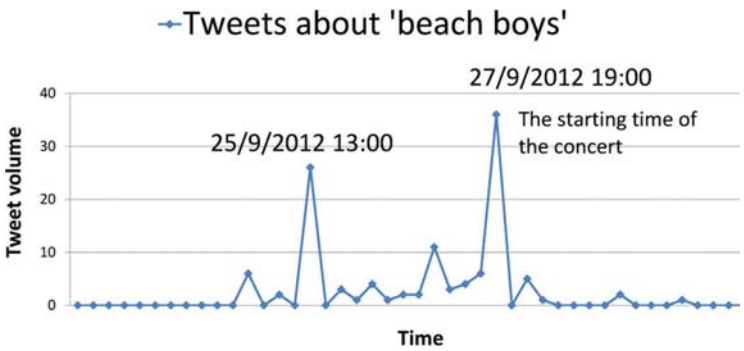

Figure 3.5 A plot of the volume of tweets in London that contain the phrase "beach boys" over time.

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 [10, 54] and (ii) detecting *unknown* events using user-generated content in social media [11, 45, 55]. 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 [55], or simply clustering social media content based on a database of targeted events [11], 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 [45].

Unlike [45], 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 [45]. 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.[6]

The work presented here advances the state-of-the-art in detecting and locating unknown events in social media and proposes a new IR task of local event retrieval, which is described next.

3.3.2 Problem Formulation

Our overall goal 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 new event detection broadcast news task of the TDT (Topic Detection and Tracking) evaluation forum.

[6]Treating social media as a social sensor has also been suggested in previous work, for example [54] and the EU FP7 social sensors project http://www.socialsensor.eu

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 $\mathcal{L} = \{l_1, l_2, \ldots\}$ 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. Examples of events to retrieve 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").

When using Twitter as a social sensor, a location l_i at a certain time t_j is characterised by the tweeting activities observed at that location within a given timeframe $(t_j - t_{j-1})$. The tweeting activities are represented with a set of tweets originating from that location shared publicly within the given timeframe $(t_j - t_{j-1})$. This set of tweets is denoted by $T_{i,j}$. Note that the fixed timeframe is defined using an arbitrary sampling rate θ; $\forall j : t_j - t_{j-1} = \theta$. 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 tweeting 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 tweeting 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.

3.3.3 A Framework for Event Retrieval

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 3.3.2, as a first step, we are not aiming to determine the finishing time of an event. As discussed in Section 3.3.2, here 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 tweeting activity [66] 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 tweeting 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 timeframe 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:

$$R(q, \langle l_i, t_j \rangle) \propto (1 - \lambda) \cdot S(q, T_{i,j}) + \lambda \cdot E(q, \langle l_i, t_j \rangle) \tag{3.1}$$

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 combination in Equation (3.1). Next, we show how we approach the problem of quantifying each component.

Aggregating Tweets

To estimate $S(q, T_{i,j})$ in Equation (3.1), 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 [8, 41]. Similarly, the tuple $\langle l_i, t_j \rangle$ is associated with a set of tweets. Inspired by [41], the score of each tuple (candidate) can be estimated by aggregating the retrieval scores (votes) for each tweet (document) associated to it. In [41], 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:

$$S(q, T_{i,j}) = \sum_{t \in Rel(q) \cap T_{i,j}} (Score(q, t)) \tag{3.2}$$

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 [53]. Higher scores represent more topically related tweets for the considered tuple.

Change Point Analysis

The problem of quantifying the score $E(q, \langle l_i, t_j \rangle)$ in Equation (3.1) maps well to *change point analysis*, a previously studied problem in the statistics literature, e.g. [34, 35]. 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 [29]. 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 [27] 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 [47]. 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 (3.1) as follows:

$$E(q, \langle l_i, t_j \rangle) = E_c(t_j) = 1 - e^{(\frac{-\ln 2}{z} \cdot v)} \tag{3.3}$$

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 (3.3) 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 timeframe 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 above, i.e. $r_j = S(q, T_{i,j})$. We call this a *query dependent* (QD) tweeting rate.

It should be noted that this framework can operate in a real-time fashion on top of the SMART architecture (Section 3.2) where social feeds are incrementally indexed such that the retrieval components are able to provide the freshest results.

3.3.4 Summary

We have devised an event retrieval framework that is capable of identifying and ranking local events in a response to a user query. In [5], we have conducted an experiment on a large collection of geo-located tweets (over 1 million) collected during a period of 12 days within London. Aligned with the tweets, we have collected, through the use of crowdsourcing, local events that took place in London from local news sources. We have evaluated the effectiveness of our framework in identifying and ranking these events through its application on the geo-located tweets. Our empirical results suggest that detecting local events from Twitter using our framework is feasible but challenging. In particular, the results show that our event retrieval framework is capable of identifying and ranking "popular" events (those found by crowd-workers and reported in the web) within a city. However, when applied on more localised events, the retrieval effectiveness of the framework degrades, possibly because of their low coverage on Twitter. To deal with this caveat, in the next section, we fuse the metadata extracted from physical IoT sensors along with the social media activity to identify topics of events happening in the real-world.

3.4 Using Sensor Metadata Streams to Identify Topics of Local Events in the City

In Section 3.3, we addressed local event retrieval by using social media activity as a sensor to detect and rank events. However, social media may only cover very popular events as users may not necessarily comment on all events taking place in the city. Therefore, physical sensors that record observations about the status of the environment can provide additional evidence about the events taking place in the city. These sensors can take the form of visual sensors such as CCTV cameras, acoustic sensors such as microphones or possibly environmental sensors.

There is a wealth of research on identifying low-level human activities from acoustic and visual sensors. Often, this involves sensor signal processing to extract sensor features for modelling human activities. For example, Atrey *et al.* [7] developed a Gaussian Mixture Model using a variety of features derived from audio signal processing to classify human activities into vocal classes, such as talking and shouting, and non-vocal classes, such as walking and running. Similar approaches also used audio signal features to identify low-level human activities that are related to security incidents, such as breaking glass or explosions [25]. In addition to using acoustic sensors, several studies have been conducted to identify low-level human activities from videos. Since its introduction in 2002, the TRECVID evaluation campaign [49] has tackled a variety of content-based retrieval tasks from video recordings to support video search and navigation. This includes the semantic indexing of video segments, whereby videos are mapped to concepts, which can be certain objects or human activities [49]. Another related task is multimedia event detection, where the aim is to identify predefined classes of events in the videos. In this task, the existing effective approaches employ classifiers trained with motion features from the videos [50]. Moreover, classifying human interactions identified in video recordings has been studied to detect surveillance-related incidents [18].

Although the aforementioned approaches derive useful semantics about the multimedia content, they only consider low-level human activities. In other words, they provide sensor metadata describing low-level human activities in the physical environment. However, to the best of our knowledge, no previous work has investigated combining these sensor metadata to detect and retrieve higher level complex events taking place in the city, such as music concerts or entertainment shows, which may involve several lower

level human actions. Here, we propose an approach for combining sensor metadata streams to support local event retrieval. We devise a supervised machine learning approach that combines sensor metadata to identify the topic of a potential event happening at a particular time in a certain location of the city. The topic corresponds to a set of terms representing a type of events, such as a concert or a protest. Our approach uses features from acoustic, visual and social sensor metadata. We also incorporate background features from past observations to model events that exhibit cyclic patterns such as traffic jams at peak times.

In Section 3.4.1, we define the problem of event topic identification that we tackle. This work makes use of sensor observations that are described in Section 3.4.2 – i.e. analysed video and audio recordings from two vibrant locations in the centre of a major Spanish city over a period of two weeks. Then to address the event topic identification problem, we discuss a supervised approach with two steps. In the first step, as described in Section 3.4.3, we obtain event annotations on the video and audio recordings dataset. In the second step (Section 3.4.4), we use the obtained annotations to map typical events taking place in those locations into coherent topics using a topic modelling technique.

3.4.1 Definition of Event Topic Identification Problem

The aim here is to combine the sensor metadata observations captured at different locations in a city to identify topics of potential high-level events taking place within certain locations. Formally, for a location l_i in a city, we denote the sensor metadata observations captured at time t_j in that location l_i by the vector $\overrightarrow{N}_{\langle l_i,t_j\rangle}$. The sensor metadata observations may include the crowd density identified from captured videos in the location, low-level audio events identified from the acoustic sensors installed in the location or social media activities, such as tweets posted by people at the location. The problem of event topic identification is to use the vector $\overrightarrow{N}_{\langle l_i,t_j\rangle}$ to map the tuple of time and location $\langle l_i, t_j\rangle$ to a certain topic $p_x \in \mathcal{P}$ described by a set of terms T_x; where $\mathcal{P}$ is a set of predefined topics.

In the previous section, the textual content of public tweets has been used as the only source of sensor metadata observation to identify topics of local events. Although this has worked well on popular events that attract social media activities, it does not work as well on more localised events that may not attract coverage on social media [5]. To alleviate this shortcoming, we introduce *physical* sensor metadata streams that can provide an additional

evidence for the topic of an event, namely video and audio metadata observations. However, this requires understanding the semantics of visual scenes or audio recordings, which remains an open challenge. Indeed, there is no known taxonomy that maps sensor metadata to topics of high level events. To address this challenge, we propose to learn the topic associated with a tuple $\langle l_i, t_j \rangle$ from *labelled* training data using features extracted from the sensor observations $\overrightarrow{N}_{\langle l_i, t_j \rangle}$.

For this purpose, and to collect labelled training data, we obtain event annotations on a pool of videos that are identified as potential candidates to contain events. Furthermore, to extract a predefined set of coherent event topics, we apply topic modelling on the descriptions of the annotated events. We detail the event annotation and the topic extraction in Section 3.4.3. Next, we describe the sensor data collection.

3.4.2 Sensor Data Collection

Our study considers two locations in the city centre of Santander in Spain. The first location is the geographical and business heart of the city; it is a major square opposite to the municipality building. The second location is a popular open market in the city, where hundreds of people go every day for shopping, located behind the municipality building. Both locations represent vibrant and busy areas, where we expect to observe high-level events of interest such as music concerts, entertainment shows or even protests. The data collection occurred during October 2013 in both locations using an edge node deployed in Santander (see Section 3.2.1).

Table 3.1 provides a summary of the sensor data collection and the metadata produced by processing the output from the microphones and the camera in each location. For producing the audio metadata, a supervised classifier using feed forward multilayer perceptron network and low-level audio features, such as those described in [20], was developed for each of the following 6 audio classes described in Table 3.1: “crowd”, “traffic, “music”, “applause”, “speaker”, and “siren”. For video metadata, the video was processed for crowd analysis where we calculate the crowd density, in desired areas, by estimating the foreground components of the video. In addition to the acoustic and visual sensors, we collected parallel social media activity in the city. In particular, using the Twitter Public Streaming API[7], we obtained tweets related to each location (as identified by their geo-locations).

[7] https://dev.twitter.com/streaming/public

Table 3.1 Summary of sensor data collection

Locations	2 (square & market)
Physical sensors	A camera and microphones (in each location)
Raw output	1600 × 1200 video @ 20 fps 16 Khz audio @ 64 kbits/s (audio is multiplexed with the video)
Audio metadata	classification scores for 6 audio classes (i) "crowd": noise from a crowd of people (ii) "traffic": car and road noises (iii) "music": music played outdoors (iv) "applause": applause, yelling or cheering (v) "speaker": speech over loud speakers (vi) "siren": noise of police cars & ambulances
Video metadata	crowd density in the scene
Twitter	geo-tagged tweets within each location

3.4.3 Event Pooling and Annotation

In this section, we describe our approach for obtaining event annotations on the recordings collected from the two used locations. Recall that our ranking units (the documents) are tuples of time and location. Each tuple represents a segment of recordings at a location. The length of the segment, the sampling rate to obtain the tuples, can be predfined and we follow [5] in setting the sampling rate to 15 minutes. Coarser- or finer-grained sampling rate can be investigated in future work for different types of events e.g. emergency events may require a finer-grained sampling.

For annotation, we consider a period of 2 weeks starting from 19 October 2013, around a week after the start of the data collection (11 October 2013) to allow the estimation of background features. Since it is expensive to examine all recordings and annotate them with events, we employ a pooling approach [16], as commonly used in IR (Information Retrieval) evaluation forums, such as TREC. For pooling, we identify candidate segments of videos where high-level events may have occurred by applying the change component of the event retrieval framework described previously (c.f. Section 3.3). In particular, the change component of this framework identifies segments where sensor metadata observations change unusually in a location, e.g. unusual change in crowd density. We use 4 different types of sensor metadata observations to generate the pool (a subset of those listed in Table 3.1): (i) the median values of the video crowd density, (ii) the median values of the crowd audio classification score, (iii) the median values of the music audio classification

score, and (iv) the total number of tweets posted. As a result, we obtain a total of 155 candidate segments. The video recording software produced videos with lengths of either 30 minutes or 1 hour, and the total number of video recordings that correspond to the 155 segments are 69 videos.

The generated candidate segments of videos were then annotated by two groups of human annotators, English and Spanish annotators, who were asked to examine the videos, describe events that they observe by typing in terms, and rate their intensity on a 3-point scale (Low, Medium, and High) according to how likely they are to generate public interest. The intensity is akin to graded relevance used in traditional IR evaluation approaches [59]. We provided the annotators with a web-based interface, of which we show a snapshot in Figure 3.6.

Statistics of the obtained annotations are summarised in Table 3.2. From the last row it can be observed that we obtain a total of 55 annotated videos, of which 21 were annotated by more than one annotator. The agreement between annotators is estimated by converting the intensity levels to binary decisions,

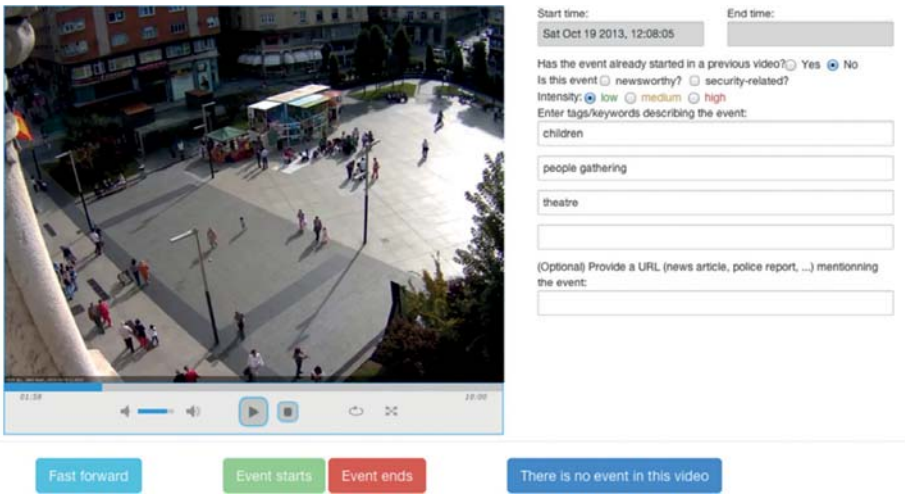

Figure 3.6 Components a snapshot from the annotation interface.

Table 3.2 Statistics from annotated videos

Annotators	Unique Videos	Annot. Ratio	Mutliple Annotations	Agreement
4 English	29	29/69 = 42%	1 video	100%
5 Spanish	47	47/69 = 68%	13 videos	77%
Both	55	55/69 = 79%	21 videos	71%

using "Medium" as a threshold. We observe that a reasonable agreement is achieved in all cases (lowest is 71%), which gives us confidence in the annotations obtained.

For the set of annotated pooled segments, we obtain terms describing events that were identified in these segments. For each annotated segment, we construct a virtual document that consists of all of the terms provided by the annotators. Since the pooled videos were annotated by both Spanish and English annotators, these virtual documents are bilingual and contain English and/or Spanish terms. Figure 3.7 presents word clouds for frequent terms occurring in the English (a) and Spanish (b) annotations, respectively, where word size is indicative of frequency in the annotations.

To cluster events into various topics, we propose to use topic modelling on the document collection of all constructed virtual documents of terms. We use the Latent Dirichlet Allocation (LDA) topic modelling implemented in the Mallet toolkit [43]. In Table 3.3, we list the top terms of 7 identified topics from the English annotations only. From the table, we can observe that the identified topics are reasonable where we see some interesting associations of terms that describe typical high-level events taking place in the square and the market, e.g. 'demonstration' and 'show' in topic 4, and 'children' and 'entertainment' in topic 6.

3.4.4 Learning Event Topics

In this section, we discuss our supervised approach for event topic identification, where the aim is to identify the topic of a segment $\langle l_i, t_j \rangle$ using the sensor metadata observations $\overrightarrow{N}_{\langle l_i, t_j \rangle}$. To train our supervised approach, we construct a *labelled* dataset of event topics from the annotated video pool collected. The labelled data consists of segments (tuples of time and location) labelled with

(a) English. (b) Spanish.

Figure 3.7 Word clouds for frequent terms occurring in the English (a) and Spanish (b) annotations, respectively, where word size is indicative of frequency in the annotations.

Table 3.3 Topics identifed with topic modelling using the english annotations

Topic	Top Terms of the Topic
Topic 1	loudspeaker people fanfare police drums procession
Topic 2	microphone rings speech public claps
Topic 3	gathering plaza people booth theatre music
Topic 4	demonstration sitting event sound speak show
Topic 5	market protest cars ongoing children fair
Topic 6	children people shopping middle entertainment
Topic 7	music singing playing guy bells whistles

either an event topic or with the label 'no event' indicating that no event of interest has occurred in the corresponding time and location. We labelled each annotated segment in the pool to the most probable topic according to the LDA topic modelling configured by setting the number of topics to 7.[8] Unlabelled segments or where the annotators did not identify any event are associated to the 'no event' label. To illustrate the volume of the data and the distribution of labels, we detail in Table 3.4 the number of segments for each label when using topic modelling on all Spanish and English annotations and setting the number of topics to 7.

We consider the problem of identifying the topic of a pooled segment as a classification task. Using the constructed labelled data, we train a binary classifier for each of the labels with features derived from various sensor metadata streams. Our intuition is that such labelled data would allow us to learn the semantics of a combination of sensor metadata. In other words we aim to match sensor metadata to topics defined using the annotations. For training the classifier, we investigate two main sets of features for the segment, *observation features* and *background features*. Table 3.5 summarises those features. The *observation features* are extracted from the sensor metadata observed in the location and time corresponding to the segment. The *background features* aim to model past observations and cyclic patterns of activities that take place over time in the same location. The intuition is that some events

Table 3.4 Distribution of labels

Lab.	#	Lab.	#	Lab.	#	Lab.	#
top.1	12	top.3	2	top.5	0	top.7	11
top.2	8	top.4	32	top.6	24	no event	66

[8]We use 7 topics since we have observed that with this setting we obtain the most coherent topics after experimenting with other alternatives (varying the number of topics between 5 and 10).

Table 3.5 Features devised for topics identification

8 Observation Features		
Audio features	6	median of the classification score for each audio class (crowd, traffic, music, applause, speaker, siren)
Video features	1	median of the crowd density score
Twitter features	1	number of tweets geotagged within the location in the past one hour
16 Background Features		
Daily aggregates	8	for each of our 8 observation features its daily median from all available past observations at the same time from previous days
Weekly aggregates	8	for each of our 8 observation features its median from all available past observations at the same time on the same day of previous weeks
Total	24	

are periodic and exhibit a long-term pattern, e.g. traffic jams at peak times resulting in a high traffic audio classification score, or entertainment shows taking place in the square at the same time on the weekends. Modelling cyclic patterns, i.e. daily and weekly cycles, from the sensor metadata observations would enable the supervised classifier to identify recurring background events or noise which are not of interest such as traffic jams. Similarly, it would help to identify recurring events of interest such as entertainment shows.

Using the labelled dataset of segments along with the features described in Table 3.5, we apply supervised machine learning to learn a binary classifier for each label. In particular, we experiment with Random Forests [15] as a learning algorithm.[9] Next we conduct a number of experiments to evaluate the accuracy of our classifier and the effectiveness of the various devised features.

3.4.5 Experiments

To evaluate our approach for identifying the topic of a candidate segment, we use the dataset of labelled segments described in Section 3.4.4. We perform a 10-fold cross validation and report the average accuracy across all labels (a label for each topic and the label 'no event'). In addition to using different instantiations of our classifier, we also compare our classifier to an alternative baseline. The "majority" baseline assigns the most common label in the

[9]We also experimented with other supervised machine learning algorithms, such as naive Bayes and SVM, however we only report results with Random Forests since they achieve the best performances and the conclusions with other algorithms are similar.

training data to the segments in the testing data. Table 3.6 summarises the results.

We observe from the table that all instantiations of our approach are markedly better than the majority baseline. In particular, when using only the observation features our approach achieves an F_1 accuracy of 0.686. We also observe that this performance further increases when using the background features. Indeed the best performance is achieved when using all background features along with the observation features ($F_1 = 0.766$). This illustrates that modelling cyclic patterns by aggregating sensor metadata from previous observations helps in better identifying whether a candidate segment represents an event and in identifying the topic of an event.

Furthermore, we conduct an ablation study to identify which features are more effective for topic identification. We remove one of our 8 observation features when learning the classifier. We report the results in Table 3.7. For example, the row headed "– (Audio crowd)" means that we use all the observation features apart from the audio crowd score. We observe that removing any of the features results in a degradation of performance for accuracy and precision. This is an interesting observation and highlights the importance of having rich metadata describing the environment for identifying the topics of high-level events. However, we also observe that the performance

Table 3.6 Performance of topic identification

Approach	F_1 Accuracy	Precision	Recall
Majority baseline	0.254	0.181	0.426
Obs. Feat.	0.686	0.705	0.697
Obs. & Daily	0.740	0.759	0.761
Obs. & Weekly	0.715	0.715	0.729
Obs. & All background	**0.766**	**0.781**	**0.762**

Table 3.7 Results of the ablation study

Model	F_1 Accuracy	Precision	Recall
All observation features	**0.686**	**0.705**	**0.697**
– (Audio crowd)	0.635	0.624	0.635
– (Audio traffic)	0.681	0.678	0.691
– (Audio applause)	0.680	0.678	**0.697**
– (Audio music)	0.685	0.682	**0.697**
– (Audio speaker)	0.657	0.656	0.665
– (Audio siren)	0.656	0.655	0.665
– (Video crowd)	0.652	0.651	0.665
– Twitter	0.682	0.677	0.697

degrades most when removing the audio crowd score and the crowd density features. This suggests that the crowd level, as detected by the acoustic or visual sensors, is important to identify events and to distinguish their topic.

3.4.6 Summary

In summary, we described an approach for fusing sensor metadata streams to identify the topics of events, happening within a city, building on the SMART framework (described previously in Section 3.2). In particular, this approach trains a classifier to identify event topics from candidate segments of audio and video recordings. Experimental results demonstrate that the best accuracy for event topic identification can be achieved by combining features from a variety of diverse sensors (acoustic, visual and social). This shows the advantages "that the social and other IoT stream fusion brings to event topic identification. Moreover, it paves the way towards more effective local event retrieval that harness both physical and social sensor streams in cities with significant IoT-connected sensing infrastructures, by combining the visions from Section 3.2 (an infrastructure for searching IoT), Section 3.3 (location event retrieval from social streams) and event topic identification from physical and social sensor streams.

3.5 Venue Recommendation

The advances of smartphone devices and wireless communication technologies have enabled people to search for information in almost every situation, and no longer simply when at a desk. However, as the information on the internet has dramatically grown every day, searching for relevant information seems to be a difficult and time-consuming task for instance, due to the cognitive complexities of expressing information needs by typing on a smartphone screen. For this reason, recommendation systems have become ubiquitous tools to obtain information, by predicting what the user wants without the need for an explicit query.

In recent years, Location-based Social Networks (LBSNs) have emerged, such as Foursquare[10] and Yelp[11], which enable users to search for Points-of-Interest (POI) or venues[12], share their physical location (check-ins[13]) as well

[10] https://foursquare.com/

[11] http://www.yelp.co.uk

[12] We use the term venue or POI interchangably.

[13] A term used by Foursquare to denote users sharing their current location with the LBSN.

as rate and comment after visiting a POI. Moreover, other users may consider those ratings and comments to select the POIs to visit at a later time. The recommendation of appropriate POIs to users, e.g. a restaurant they are likely to visit, has become an important feature for LBSNs, which assists people to explore new places and helps business owners to be discovered by potential customers.

Venue recommendation is an example of a recommendation task: given no explicit 'query' by the user, but knowledge about the user's preferences and about the venues, can a system predict which venues the user may wish to visit? The types of information that are available for this task are summarised in Table 3.8. For instance, if the user has checked-in or rated other venues before, then this provides user preference information, which can be used by collaborative filtering approaches to suggest venues of interest (discussed in details in Section 3.5.1 below).

Nevertheless, information about each venue itself can help to predict its likely suitability. For instance, the Foursquare website lists a city park as a top nearby venue, regardless of the time of day, when (say) late in the evening that park may be both closed and unsuitable to visit. Therefore IoT and social sensing technology have a role to play in predicting the occupancy of venues. Predicted occupancy and similar measures of popularity are examples of venue-dependent features (i.e. which are the same for all users) – and are discussed further in Section 3.5.2.

Finally, the *contextual* situation of the users when requesting venue recommendations can also have an impact on the appropriate choice of venues: clearly, context can encapsulate the location of the user – as nearby venues are more likely to be useful to the user; however, the people they are with (alone, with colleagues, family or friends) may also significantly impact upon the most appropriate choice of venues. In Section 3.5.3 we highlight recent work in context-aware venue recommendation.

In the remainder of this section, we discuss recent research in venue recommendation, particularly highlighting our own work, which builds upon

Table 3.8 Sources of data for venue recommendation

	Example Sensors	
Information Type	Physical	Social
Venue	# cell phones nearby	# recent check-ins
	# subway exists nearby	# comments
User Preference	User's distance to the venue	The user likes similar venues
		The user has commented positively about a similar venue

the SMART architecture (described in Section 3.2). The three different sources of evidence, and how they are modelled, are described in turn: user preferences (e.g. through collaborative filtering approaches) (Section 3.5.1), venue dependent evidence (Section 3.5.2), and contextual preferences (Section 3.5.3).

3.5.1 Modelling User Preferences

In terms of modelling the preferences of users, collaborative filtering (CF) is a widely used technique to generate personalised recommendations. CF typically exploits a matrix of user-venue preferences in order to generate venue recommendations for individual users. There are two major categories of traditional CF approaches namely memory-based CF and model-based CF [1, 26]. The memory-based CF approaches are categorised as user-based or venue-based. A typical user-based CF approach predicts a user's rating on a target venue by aggregating the ratings of K similar users who have previously rated the target venue. The similarity between two users is usually identified using the Pearson correlation or the cosine similarity upon their rating vectors [57]. Intuitively, the user-based CF approaches assume that users who share similar preferences will like the same venue e.g. I like what my friends like. The extension of user-based CF approaches has been shown to improve the quality of recommended venues such as through the introduction of fine-grained neighbour-weighting factors [32] or by exploiting a recursive neighbour-seeking scheme [65]. In contrast, venue-based CF approaches suggest venues on the basis of information about other venues that a user has previously rated [21]. The suggestion of venues for a given user are ranked by aggregating the similarities between each candidate venue and the venues that the user has rated. Although typical memory-based CF approaches have been shown to be effective in suggesting venues to users, the main drawback of such approaches is that the computation of similarities between all pairs of users or venues is expensive due to its quadratic time complexity. Moreover, as memory-based CF approaches are dependent on the availability of human ratings, the effectiveness of these approaches significantly decreases when they are faced with sparse ratings.

On the other hand, model-based CF approaches were introduced to address the shortcomings of memory-based CF approaches [14, 9]. Such approaches are based on supervised models which are trained on the user-venue matrix [1, 26]. The trained prediction models can then be used to generate suggestions for individual users. Recently, the most well-known technique of model-based CF approaches is matrix factorization (MF) [36]. The advantages of

MF techniques are their scalability and accuracy. Generally, MF models learn latent features of users and venues from the information in the user-venue matrix, which are further used to predict new ratings between users and venues.

Various recent works on venue recommendation have exploited user-generated information (e.g. check-in and venue information) from LBSNs. Such approaches typically apply widely-used CF techniques to suggest personalised venues to users. For example, friend-based CF approaches can recommends POIs to visit based on collaborative ratings of venues made by the user's friends [40, 62]. Yang *et al.* [61] proposed a model that estimates a venue's quality based on a sentiment score calculated based on the tips (comments) made by users in the LBSN, and then recommended venues based on this sentiment score.

Even if there is no information available about which venues a user has previously visited, other proxy information can still be obtained with which to personalise the suggestion of venues. For instance, one venue recommendation approach that we have proposed in [23] used the users' Facebook profiles to permit personalisation, even when the venues being suggested were from the separate Foursquare LBSN. In general, we use a probabilistic model to describe the preferences of a user determined from Facebook – as well as the likely interests of users – in terms of a coarse-grained ontology from the Open Directory Project (DMOZ.org): e.g. Arts, Games, Health, Technology etc. In particular, we examine the entities liked by users on Facebook to build up a preference distribution over the DMOZ categories. However, as the Foursquare entities may only have a single name to describe them, this may be insufficient information to accurately predict which DMOZ categor(ies) these entities should belong to. To alleviate this problem, we issue each Facebook entity's name as a query to a web search engine, and analyse the contents of the returned pages to determine which categories they belong to. This allows a model of the users' preferences to be determined based on their Facebook' Likes.

Similarly, when considering a venue, we determine which categories that it should belong to by also issuing the name of the venue to a web search engine. Figures 3.8 (a) and (b) pictorially depict the aforementioned processes for the Facebook' Likes and the venues, respectively. Then, the personalisation of venues suggested to a user can be achieved by suggesting venues with more similar category distributions to that of the users. Through a user study involving 100 users and three different cities (Amsterdam, London, San Francisco), we evaluated our complete probabilistic model [23]. Our findings suggested that while our personalised model was effective, it was residents

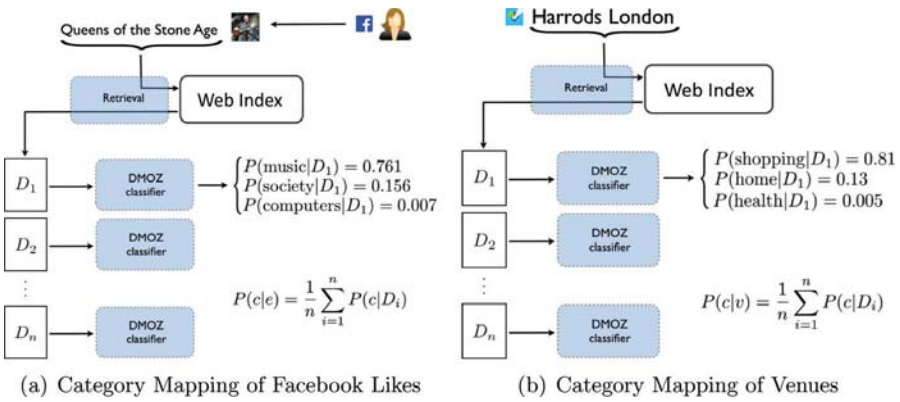

Figure 3.8 Obtaining DMOZ category distributions from Facebook' likes and venues.

rather than visitors to a city who were found to most prefer personalised venue suggestions.

Overall, we have shown how personalised venue suggestions can be achieved, but there are other factors that may cause a venue to be selected by a user or not – indeed, our own user study in [23] found that tourists were more attracted by popular venues, particularly during the evening. In the next section, we describe how measures of the venue's popularity can be sensed, both physically or socially.

3.5.2 Venue-dependent Evidence

The act of checking-in on a LBSN such as Foursquare provides a number of signals about a venue – the short-term popularity of a venue, as well as an aggregate signal about its popularity at this time, as well as at the current day of the week and season of the year. IoT-connected sensors that can detect a busy venue, such as through CCTV analysis and/or audio analysis (see Table 3.8) can also similarly be used.

To predict the attendance of a venue, we constructed a time series of attendance for each individual venue [23]. Time series are numerical information that are observed sequentially over time. By obtaining the number of people currently visiting the venue from Foursquare every hour for each venue, we are able to build a comprehensive time series of venue attendance. Figure 3.9 shows how a state-of-the-art neural network-based approach can predict the attendance of the famous Harrods department store in Knightsbridge, London. Using such predicted occupancy figures for venues improves the effectiveness of a venue suggestion approach [23] – as illustrated in Table 3.9.

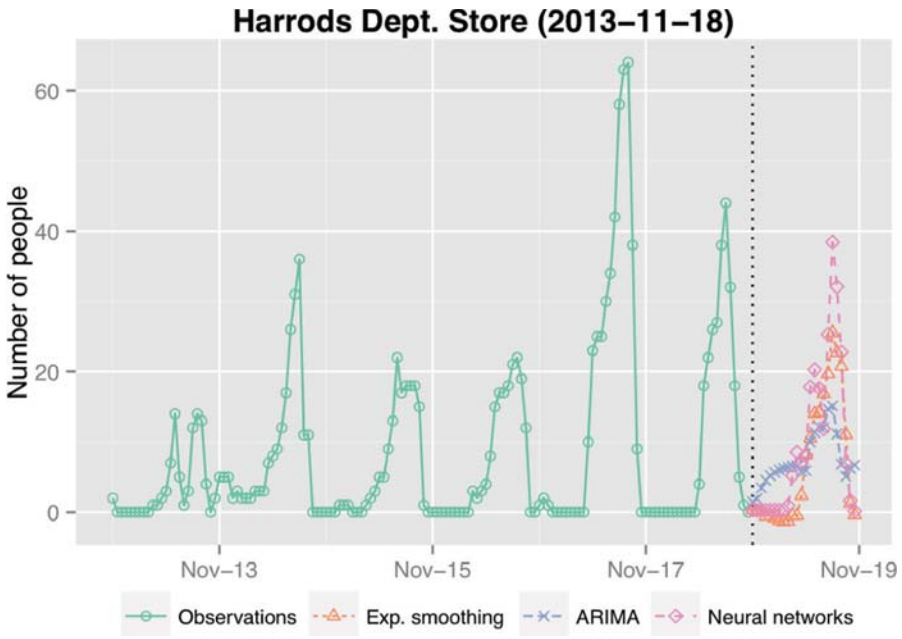

Figure 3.9 Predicting occupancy from Foursquare check-in time-series.

Table 3.9 Examples of venue recommendations produced by our model for user in a central location in London at two different times

Friday 03 April 14:00	Sunday 5 April 00:00
Debenhams	Novikov Restaurant & Bar
Natural History Museum	Boujis (nightclub)
Selfridges & Co	
National Gallery	
Apple Store	
London Victoria Railway Station	
Victoria and Albert Museum (V&A)	
Millbank Tower	
Science Museum	
Piccadilly Circus	

Of course, there are other sources of evidence in a LBSN that are indicative of a venue's popularity and hence a priori its suitability for recommendation to any user. In a separate work, we examined a number of venue-dependent features for making effective venue suggestions, such as the number of check-ins, number of photos, average ratings etc. [22] (summarised in Table 3.10).

To create an effective venue suggestion model, these features were combined with user-venue features (which model the user's venue preferences, in order to make personalised suggestions) within a learning-to-rank approach. In doing so, the application of the learning-to-rank approach [22] aims to find a combination of the features that can best satisfy users, determined by a set of training observations (users with known venue preferences). The experiment made use of the state-of-the-art LambdaMART learning-to-rank approach [60], which is an adaptation of gradient boosted regression trees to make effective rankings.

Figure 3.10 shows how performance can increase or decrease when venue-dependent features are removed from the model. This takes the form of an *ablation* experiment to explore the individual effectiveness of these venue-dependent features, in order to determine which single features are the most effective when suggesting venues to users. In this experiment, we consider the LambdaMART ranking model – learned using all features – as a baseline, and we compare its performances to other LambdaMART models that have been learned after removing each of the venue-dependent features individually – a decrease in performance implies that the feature is deemed useful.

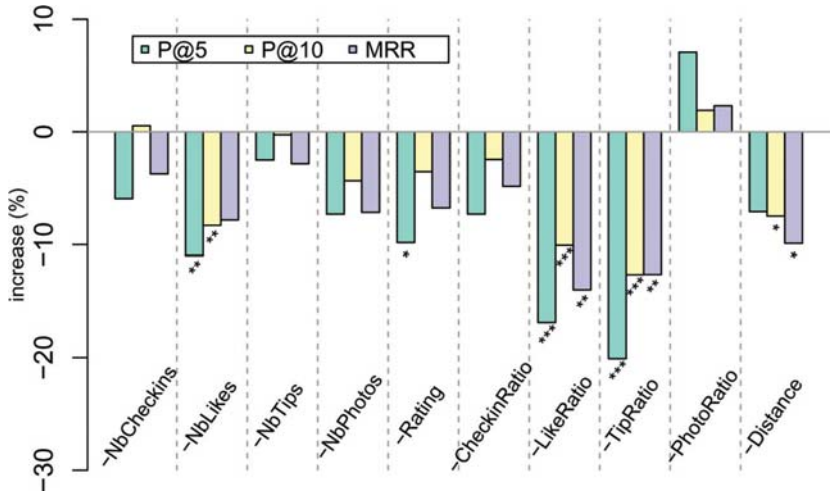

Figure 3.10 Percentage of improvement obtained when independently removing single venue-dependent features, with respect to a LambdaMART baseline that uses a total of 64 features. Improvements are expressed in terms of P@5, P@10, and MRR. Statistical significance is stated according to a paired t-test (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$). *Source*: [22].

Table 3.10 Venue-dependent Foursquare features used by Deveaud et al. [22]

Feature Name	Description
NbCheckins	Total number of check-ins in the venue.
NbLikes	Total number of "likes" for the venue.
NbTips	Total number of "tips" (comments) for the venue.
NbPhotos	Total number of photos for the venue.
Rating	Average of all the ratings given by the users for the venue.
CheckinRatio	$\frac{\text{NbCheckins}}{\text{NbCheckinsInCity}}$
LikeRatio	$\frac{\text{NbLikes}}{\text{NbLikesInCity}}$
TipRatio	$\frac{\text{NbTips}}{\text{NbTipsInCity}}$
PhotoRatio	$\frac{\text{NbPhotos}}{\text{NbPhotosInCity}}$
Distance	Distance of the venue from the center of the city.

On analysis of Figure 3.10, the first observation we make is that PhotoRatio appears to be harmful. When Foursquare venues do not have any photo, the value of this feature is equal to zero, which seems to confuse the learner.

Likes and tips, which are more abundant and hence do not suffer from this problem, appear to be very strong indicators of relevance. It is important to note that the raw numbers (i.e. NbLikes and NbTips) are not enough, and that using the city context greatly improves the importance of these features (see LikeRatio and TipRatio). The rating of the venue (which is an average of all the ratings provided by Foursquare users) is also a good indicator of relevance, but to a lesser extent than LikeRatio and TipRatio. Finally, the distance between the venue and the center of the city also seem to play an important role. Since city centres usually are the most vibrant parts, using this distance as a feature allows the learned model to implicitly separate potentially relevant and attractive venues from unpopular ones.

Overall, our experiment in [22] – and highlighted above – shows the importance of venue-dependent features for effective venue suggestions.

3.5.3 Context-Aware Venue Recommendations

In addition to making recommendations based on user preferences and the popularity of the venue, another area that has emerged recently is context-aware venue recommendation (CAVR, also known as contextual suggestions). CAVR acknowledges that the appropriate venues to be recommended to a user may depend on the contextual environment of the user. Context as a notion is wide-reaching, but for venue recommendation, it can encompass factors detectable about the user, such as the location of the user, the time of day, the weather, as well as human factors, such as who the user is with (colleagues,

friends, partner family, etc), that they may explicitly provide to the venue recommendation system.

Various existing works have shown that considering such context and leveraging the use of user-generated data in LBSNs can significantly enhance the effectiveness of CAVR applications [37, 38, 63]. Yuan *et al.* [63] developed a collaborative *time-aware venue recommendation* approach that suggests venues to users at a specific time of the day. In particular, they leveraged the use of historical check-ins of users in LBSNs to model the temporal behaviour of the users and extend the user-based CF technique to incorporate both temporal and geographical effects using linear combination. Recently, Li *et al.* [37], proposed factorization methods for making venue recommendations, which can exploit different types of context information (e.g. the user's location and the time of the day). Previous works on CAVR (e.g. [37, 63]) used check-in data from LBSNs to evaluate the accuracy of their recommendations, by assuming that users implicitly like the venues they visited.

Since 2012, the US National Institute of Standards and Technology (NIST) have been developing reliable and reusable test collections and evaluation methodologies to measure the effectiveness of CAVR systems through the Contextual Suggestion track [19] of the Text REtrieval Conference (TREC) evaluation campaign. In particular, the task addressed by the TREC Contextual Suggestion track is as follows: given the user's preferences (ratings of venues) and context (e.g. user's location, city), produce a ranked list of venue suggestions for each user-context pair. A description of the contexts addressed in the 2015 TREC Contextual Suggestions track are presented in Table 3.11.

Table 3.11 The 12 dimensions of the contextual aspects proposed by the TREC 2015 contextual suggestion track

Aspect	Dimension	Description
Duration	Day Time	Is a venue suitable to visit between 6:00 AM – 6:00 PM?
	Night Time	Is a venue suitable to visit between 6:00 PM – 6:00 AM?
	Weekend	Is a venue suitable to visit on weekend?
Season	Spring	Is a venue suitable to visit between March and May?
	Summer	Is a venue suitable to visit between June and August?
	Autumn	Is a venue suitable to visit between September and November?
	Winter	Is a venue suitable to visit between December and February?
Group	Alone	Is a venue suitable to visit alone?
	Friends	Is a venue suitable to visit with friends?
	Family	Is a venue suitable to visit with family?
Type	Business	Is a venue suitable to visit for a business trip?
	Holiday	Is a venue suitable to visit for a holiday trip?

The availability and popularity of the TREC Contextual Suggestion track test collections has accelerated research into this challenging task. A few research groups participating in the TREC Contextual Suggestion track have attempted to explicitly model the contextual appropriateness of the venues. Hashemi *et al.* [31] applied parsimonious language models [33] to rank suggestion candidates based on the given contextual information such as trip duration and type, and information from the user's profile, such as their age and gender. Textual language models of each contextual aspect were built offline, and then the relevance of a given venue to the various contextual aspects of the user were estimated by calculating the KL-divergence of the standard language models of suggestion candidates and the language model of different contexts that was built in advance. The contextual relevance of different contextual aspects to the given suggestion candidate is trained using a pairwise SVM rank learning-to-rank model. In [44], we proposed two approaches for CAVR. First, a Factorization Machines-based approach proposed by [52] to rank the candidate venue suggestions. The factorisation machines receive as input instances that enclose the information related to a user, a venue he/she visited and the context of the visit in the form of numerical vectors. In particular, we trained the factorisation machines to reduce the error in the ranking of the user profiles by adapting the list-wise error function of ListRank [56] for their factorisation machine model. The second approach is a learning-to-rank based approach where contextual features are extracted from the user-generated data from LBSN (e.g. timestamp of comments and photo).

Recently, we proposed a supervised approach that predicts the appropriateness of venues to particular contextual aspects, by leveraging user-generated data in LBSNs such as Foursquare [42]. This approach learns a binary classifier for each dimension of three considered contextual aspects proposed by the TREC Contextual Suggestion track (see Table 3.11). A set of discriminative features are extracted from the comments, photos and website of venues. For instance, when travelling with children, the website of an appropriate restaurant may mention a children's menu; similarly, users may reminisce about pleasant times they had with their family using the LBSN comment functionality. By analysing these sources of evidence, we showed in Section 4.2 that both the websites of venues and comments left by users on the LBSN could accurately predict if a venue was suitable for the various contextual aspects.

3.5.4 Summary

Venue recommendation is an important task, for instance exploring a new city, as evidenced by the popularity of LBSNs and other websites such as

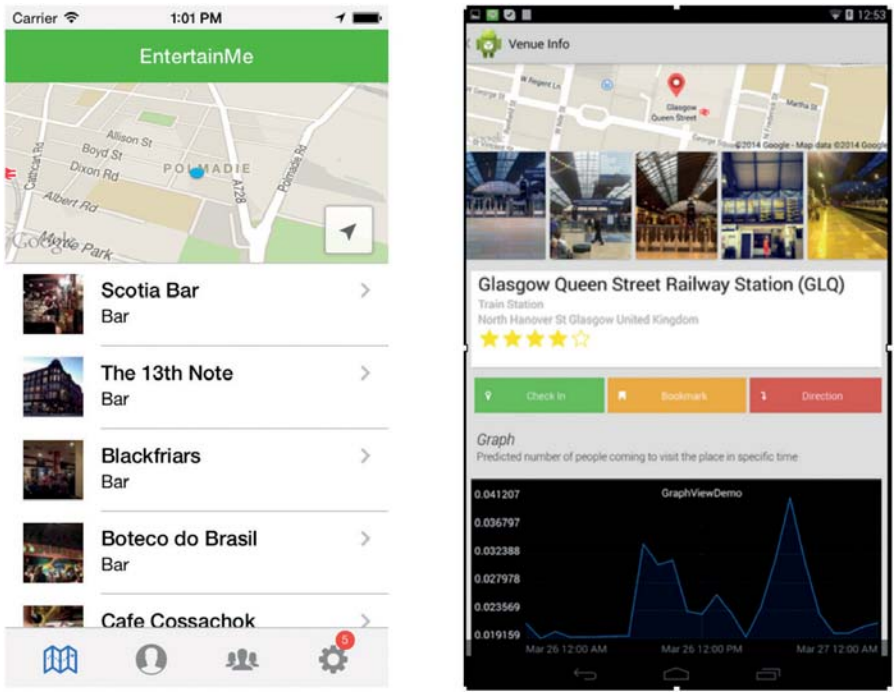

Figure 3.11 Screenshots of the EntertainMe! mobile venue recommendation application.

TripAdvisor. There is a significant body of research on venue recommendation, with new models making increasing use of social sensing. As highlighted in Table 3.8, physical sensors may assist in making recommendations, for instance, by recognising that some venues are not appropriate for poor weather conditions.

In [24], we developed a mobile application called EntertainMe!, based on the SMART architecture and employing some of the techniques described in Sections 3.5.1, 3.5.2 and 3.5.3. A screenshot of the mobile application is shown in Figure 3.11.

3.6 Conclusions

In this chapter, we described the SMART architecture, allowing to develop real-time search applications on the so-called Internet of Things (IoT) infrastructure. We illustrated the use of the SMART architecture through three applications, addressing local search, event topic identification and venue recommendation, respectively. In general, these applications show how

real-time information needs by users can be better served through the integration and fusion of IoT sensing streams with social content within a unified platform.

Over the next few years, as small IoT-enabled devices become ever more ubiquitous, search platforms like SMART will become increasingly more important as a means to convert the data collected by these sensors into useful actionable information. Indeed, we expect that the continuing adoption of IoT sensors will enable a wide variety of new user information needs to be satisfied, both in the short and long term. For instance, large shopping malls are installing IoT sensing infrastructures, creating 'SMART buildings', with the aim of enhancing the user's shopping experience [51]. This allows users' positions to be tracked as they move around the mall, which could enable better personalised search results, e.g. if a user enters a query such as 'sports shirts' into their mobile then the shopping results could be augmented with the location of those products nearby in the mall. Furthermore, in the smart utility space, devices such as smart fridges can make use of platforms like SMART to suggest contextual queries/reminders to the user. For example, if the fridge detects that the user is about to run out of milk, the fridge could push search results for milk to the user on their smart phone.

The SMART (http://www.smartfp7.eu/) platform is an open source project, intended to facilitate harnessing the power of the Internet of Things infrastructure in search applications. Source code and example applications can be downloaded from https://github.com/SmartSearch.

Acknowledgements

The authors acknowledge the support of the EC co-funded project SMART (FP7-287583).

References

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6): 734–749, 2005.

[2] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau. Sentiment analysis of twitter data. In *Proceedings of LSM '11*, pages 30–38, Portland, Oregon, USA, 2011.

[3] D. Albakour, C. Macdonald, and I. Ounis. Query scoring and anticipation subsystem. Technical Report D5.3b, SMART-FP7.eu, 2015.

[4] M. Albakour, C. Macdonald, I. Ounis, A. Pnevmatikakis, and J. Soldatos. Smart: An open source framework for searching the physical world. In *Proceedings of the SIGIR'12 workshop on Open Source Information Retrieval*, pages 48–51, 2012.

[5] M.-D. Albakour, C. Macdonald, and I. Ounis. Identifying local events by using microblogs as social sensors. In *Proc. of OAIR'13*.

[6] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking. pilot study final report. In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, 1998, pp. 194–218.

[7] P. K. Atrey, M. Maddage, and M. S. Kankanhalli. Audio based event detection for multimedia surveillance. In *Proc. of ICASSP'06*.

[8] K. Balog, L. Azzopardi, and M. de Rijke. A language modelling framework for expert *Information Processing & Management*, 45(1):1–19, 2009.

[9] C. Basu, H. Hirsh, W. Cohen, et al. Recommendation as classification: Using social and content-based information in recommendation. In *AAAI/IAAI*, pages 714–720, 1998.

[10] H. Becker, D. Iter, M. Naaman, and L. Gravano. Identifying content for planned events across social media sites. In *Proceedings of WSDM '12*, pages 533–542, New York, NY, USA, 2012. ACM.

[11] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of WSDM'10*, pages 291–300, 2010.

[12] E. Borden. Pachube internet of things "bill of rights". http://blog.cosm.com/2011/03/pachube-internet-of-things-bill-of.html, 2011.

[13] M. Botts, G. Percivall, C. Reed, and J. Davidson. Ogc sensor web enablement: Overview and high level architecture. In *GeoSensor Networks*, pages 175–190, 2008.

[14] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[15] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[16] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *Proc. of SIGIR'04*.

[17] J. Camp, J. Robinson, C. Steger, and E. Knightly. Measurement driven deployment of a two-tier urban mesh access network. In *Proceedings of MobiSys '06*, pages 96–109, 2006.

[18] F. Chen and W. Wang. Activity recognition through multi-scale dynamic bayesian network. In *Proc. of VSMM'10*.

[19] A. Dean-Hall, C. L. Clarke, J. Kamps, P. Thomas, N. Simone, and E. Voorhees. Overview of the trec 2013 contextual suggestion track. Technical report, DTIC Document, 2013.

[20] J. Dennis, Q. Yu, H. Tang, H. D. Tran, and H. Li. Temporal coding of local spectrogram features for robust sound recognition. In *Proc. of ICASSP'13*.

[21] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1): 143–177, 2004.

[22] R. Deveaud, M. Albakour, C. Macdonald, I. Ounis, et al. On the importance of venue-dependent features for learning to rank contextual suggestions. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1827–1830. ACM, 2014.

[23] R. Deveaud, M.-D. Albakour, C. Macdonald, and I. Ounis. Experiments with a venue-centric model for personalised and time-aware venue suggestion. In *CIKM 2015: 24th ACM International Conference on Information and Knowledge Management*, 2015.

[24] R. Deveaud, M.-D. Albakour, J. Manotumruksa, C. Macdonald, I. Ounis, et al. Smartvenues: Recommending popular and personalised venues in a city. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 2078–2080. ACM, 2014.

[25] A. Dufaux, L. Besacier, M. Ansorge, and F. Pellandini. Automatic sound detection and recognition for noisy environment. In *Proc. of EUSIPCO'00*.

[26] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4(2):81–173, 2011.

[27] F. Grubb. Procedures for detecting outlying observations in samples. technometrics. 11, 1969.

[28] D. Guinard and V. Trifa. Towards the web of things: Web mashups for embedded devices. In *Proceedings of WWW '09*, 2009.

[29] V. Guralnik and J. Srivastava. Event detection from time series data. In *Proceedings of SIGKDD'99*, pages 33–42, 1999.

[30] S. Hansell. Google keeps tweaking its search engine. *New York Times*, June 2007. http://www.nytimes.com/2007/06/03/business/yourmoney/03google.html

[31] S. H. Hashemi, M. Dehghani, and J. Kamps. Univ of Amsterdam at TREC 2015: Contextual suggestion track. In *Proceedings of TREC*, 2015.

[32] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM, 1999.

[33] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *Proc. of SIGIR*, pages 178–185. ACM, 2004.

[34] L. Horváth. The maximum likelihood method for testing changes in the parameters of normal observations. *The Annals of statistics*, 21(2):671–680, 1993.

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

[36] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

[37] X. Li, G. Cong, X.-L. Li, T.-A. N. Pham, and S. Krishnaswamy. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 433–442. ACM, 2015.

[38] K. H. Lim. Recommending tours and places-of-interest based on user interests from geo-tagged photos. In *Proceedings of the 2015 ACM SIGMOD on PhD Symposium*, pages 33–38. ACM, 2015.

[39] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, Mar. 2009.

[40] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 203–210. ACM, 2009.

[41] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *Proceedings of the 15th ACM international conference on Information and knowledge*

management, CIKM '06, pages 387–396, New York, NY, USA, 2006. ACM.

[42] J. Manotumruksa, C. Macdonald, and I. Ounis. Predicting contextually appropriate venues in location-based social networks. In Proceedings of CLEF 2016.

[43] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[44] R. McCreadie, S. Vargas, C. Macdonald, I. Ounis, S. Mackie, J. Manotumruksa, and G. McDonald. Univ of Glasgow at TREC 2015: Experiments with Terrier in contextual suggestion, temporal summarisation and dynamic domain tracks. In *Proc. of TREC*, 2015.

[45] D. Metzler, C. Cai, and E. H. Hovy. Structured event retrieval over microblog archives. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings*, June 3–8, 2012, Montréal, *Canada*, pages 646–655, 2012.

[46] D. O'Byrne, R. Brennan, and D. O'Sullivan. Implementing the draft w3c semantic sensor network ontology. In *Proceedings of PERCOM Workshops*, pages 196–201, 2010.

[47] M. Osborne, S. Petrovic, R. McCreadie, C. Macdonald, and I. Ounis. Bieber no more: First story detection using twitter and wikipedia. In *Proceedings of the SIGIR'12 Workshop on Time-aware Information Access. (TAIA)*., 2012.

[48] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of ACM SIGIR-OSIR 2006*, Seattle, Washington, USA, 2006.

[49] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quénot. TRECVID 2014 – An overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proc. of TRECVID'14*.

[50] S. Phan, T. D. Ngo, V. Lam, S. Tran, D.-D. Le, D. A. Duong, and S. Satoh. Multimedia event detection using segment-based approach for motion feature. *Journal of Signal Processing Systems*, 74(1):19–31, 2014.

[51] Y. Ren, M. Tomko, F. D. Salim, K. Ong, and M. Sanderson. Analyzing web behavior in indoor retail spaces. *Journal of the Association for Information Science and Technology*, 2015.

[52] S. Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

[53] S. Robertson and H. Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Information Retrieval*, 3(4):333–389, 2009.

[54] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[55] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. Twitterstand: news in tweets. In *Proceedings of GIS'09*, pages 42–51, New York, NY, USA, 2009. ACM.

[56] Y. Shi, M. Larson, and A. Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proc. of RecSys*, pages 269–272. ACM, 2010.

[57] A. Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

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

[59] E. M. Voorhees. Evaluation by highly relevant documents. In *Proc. of SIGIR'01*, pages 74–82, 2001.

[60] Q. Wu, C. J. C. Burges, K. M. Svore, and J. Gao. Ranking, Boosting, and Model Adaptation. Technical Report MSR-TR-2008-109, Microsoft, 2008.

[61] D. Yang, D. Zhang, Z. Yu, and Z. Wang. A sentiment-enhanced personalized location recommendation system. In Proceedings of the 24th ACM *Conference on Hypertext and Social Media*, pages 119–128. ACM, 2013.

[62] M. Ye, P. Yin, and W.-C. Lee. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 458–461. ACM, 2010.

[63] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 363–372. ACM, 2013.

[64] H. Zaragoza, B. B. Cambazoglu, and R. Baeza-Yates. Web search solved?: All result rankings the same? In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management,* CIKM '10, pages 529–538, New York, NY, USA, 2010. ACM.

[65] J. Zhang and P. Pu. A recursive prediction algorithm for collaborative filtering recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 57–64. ACM, 2007.

[66] A. Zubiaga, D. Spina, V. Fresno, and R. Martínez. Classifying trending topics: a typology of conversation triggers on twitter. In *Proceedings of CIKM'11*, pages 2461–2464, New York, NY, USA, 2011. ACM.