SMART: An open source framework for searching the physical world

M-Dyaa Albakour, Craig Macdonald
Iadh Ounis
University of Glasgow, UK
{dyaa, craigm, ounis}@dcs.gla.ac.uk

Aristodemos Pnevmatikakis
John Soldatos
Athens Information Technology, Greece
{apne, jsol}@ait.gr

ABSTRACT

User queries are becoming increasingly local where people are interested in what their friends are up to or what is happening in their local area. Sensors can assist in *localised* information retrieval by giving the search engine direct access to events happening in the local world. In this paper, we describe an open source framework to search in real-time multimedia and social streams. The SMART framework offers a platform to retrieve information from both the physical world and from people interactions on social media. Examples where this framework can be useful include "smart cities" where people can have information needs such as 'what parts of the city has live music on and what do people think about those music events?'. We identify the challenges of building such a framework and the motivations behind releasing it as open source software. The open architecture of the framework brings about possibilities for extending it and deploying it in a wide variety of novel applications.

Keywords

Social search, real-time search, sensor search, smart cities

1. INTRODUCTION

The internet has grown in the last decade to connect a large number of sensing devices that can monitor the physical world such as cameras, microphone arrays, or light sensors. The number of sensors connected to the internet is magnitudes higher than the number of its users [8]. The availability of such connected sensors open opportunities to collect in real-time the status of the physical world and process this information to develop novel applications in the areas of 'smart' cities, social networking, surveillance and security. This has triggered the development of tools and techniques for searching sensor data [5, 6]. However, these methods are still largely based on the indexing and searching of previously defined (and usually textual) metadata. Indeed, while those methods exploit recent advances in sensor ontologies [10] in order to decouple the queries from the low-level details of the underlying sensors, they cannot provide effective search over arbitrary large and diverse sources of multimedia data derived from the physical world.

Moreover, with the emergence of social networks such as Twitter and Facebook, one can envisage situations where information stemming from both social and sensor networks can be combined. In fact, there is a mutual benefit from the convergence of both sensor networks and social networks. Social networks can benefit from the fact that human activity and intent can be directly derived from sensors, which obviates the needs for explicit user input. For example, Foursquare[1] uses smart phone-based GPS and mapping services to enable users to track their friends on the social network platforms. On the other hand, sensor networks could start their cooperation in a social way (i.e. based on information derived from social networks) [4]. For example, think of the query "I want a good restaurant in a place that is lively now". To answer this query, we need a system that can process information about: (i) How *lively* a location is *right now*. Audiovisual crowd analysis can answer that by providing metadata from processing the signals of the connected sensors. These signals should be processed in real-time to provide timely information about the status of the environment in various locations. (ii) How *good* the various restaurants in different areas are. This needs data from social networks (user-generated content by tagging or "liking" good places) and/or from the Linked Data cloud (e.g. restaurant critics) [7].

In this paper, we present our vision for an open source framework where information stemming from large-scale inter-connected sensors and social network streams can be indexed in real-time to facilitate searching the physical world.

2. THE SMART FRAMEWORK

The SMART[2] (Search engine for MultimediA enviRonment generated contenT) framework aims to provide an infrastructure where multimedia sensing devices in the physical world can be easily used to provide information about the status of their environments and make it available in real-time for search in combination with information from social networks. The name SMART acknowledges the vision of "smart cities".

The architecture of the SMART framework is illustrated in Figure 1, where four layers are identified. At the lowest level (physical) 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 (e.g. RDF). Examples of processing algorithms can include

[1]https://foursquare.com/
[2]http://www.smartfp7.eu

SIGIR 2012 Workshop on Open Source Information Retrieval.
August 16, 2012, Portland, Oregon, USA.

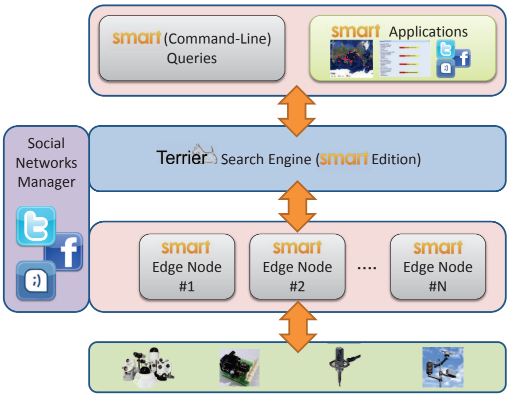

Figure 1: Architecture of the SMART framework

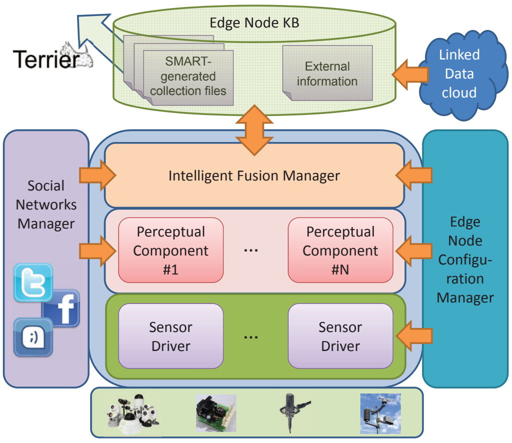

Figure 2: Edge node components

crowd data analysis for video streams and speech recognition in audio streams. The *search layer* collects the 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 identified in the sensor and social streams to rank events that are relevant to the user queries. Queries can be 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 uppermost *application/visualisation layer* offers reusable APIs to develop applications that can issue queries to the SMART search engine and process or visualise the results.

In the next sections, we will discuss in detail the top three software layers of Figure 1 and the challenges for building each layer. We then describe our vision of the open source SMART framework.

3. EDGE NODE LAYER

The edge node is introduced in this section. First, its aims and challenges are discussed, followed by an example using crowd analysis for metadata extraction.

3.1 Infrastructure Aims

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 city block or a public square in the city centre. At the edge node, the signal streams, either from physical sensors (e.g. audio/visual streams or environmental measurements), or from social networks, are processed to extract events of interest. To achieve this, the design of the edge node is influenced by: (a) state-of-the-art Internet-of-Things platforms, which typically filter, fuse, combine and reason over multiple data streams and (b) Linked Data technologies. The edge node's architecture is shown in Figure 2.

3.2 Challenges and Research Aims

The data streams collection and processing component provides a uniform way for interfacing to virtually any type of sensor and processing. These include perceptual components (i.e. the algorithms that attempt to interpret the streams' meaning and extract context) running on physical sensors' feeds (such as crowd analysis running on video feeds), as well as social networks filters implemented within the Social Networks Manager (e.g. sentiment analysis of Tweets in the local area of the sensors [1]). This component is empowered by a common unified model for the metadata of the various feeds, which alleviates their heterogeneity while also facilitating their management within the edge node. The approach is similar to that adopted by the Pachube.com[3] platform [3], yet SMART edge nodes provide support for much richer metadata. The development and adoption of a common unified metadata model for all SMART feeds ensures the openness and extensibility of the platform in terms of new sensors, perceptual components and social networking feeds. The output of this component is in the form of continuous metadata streams, which will then need to be processed in real-time. By thresholding a single continuous metadata stream, low-level events are extracted. The moment the threshold is exceeded, the low-level event is signalled as active, while the moment the stream receives a smaller value, the low-level event is signalled as inactive. The Intelligent Fusion Manager subsequently undertakes the rule-driven combination of multiple low-level events (possibly stemming from different sensors or perceptual components). This component integrates sophisticated rule engines that facilitate high-level event recognition on the basis of complex rules over multiple low-level event streams. A rule engine that leverages event calculus techniques [2] is being integrated. The reasoning component leads to a semantic description of the events (i.e. based on the RDF format), which is in line with work undertaken in the scope of semantic sensor networks [15]. The Linked Data component [7] collects data relevant to the edge node that are available as part of the Linked Data cloud. Hence, the information available to the search engine is enriched. For example, it can provide the means for identifying geo-locations [13] associated with the target events.

The edge node stores all types of metadata (the continuous metadata streams, the low-level and the high-level events) in its Knowledge Base using an XML/JSON interface that complies with the aforementioned unified data model. The edge node also provides a web service API (RESTful API) to deliver events to the search engine

3.3 Example Based on Crowd Analysis

Crowds are a main source of events for SMART, especially in a "smart city" setting where citizens are empowered with

[3] `Pachube.com is now Cosm.com`

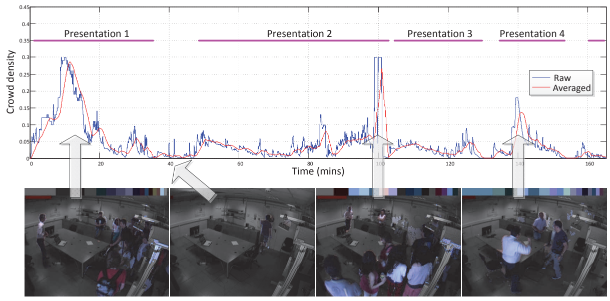

Figure 3: SMART continuous metadata and low-level crowd events as a function of time. Some typically processed frames are also shown.

knowledge about their environment. At a first level of analysis, we are interested in quantifying their density, which is based on an adaptive foreground segmentation. This segmentation is based on a variant of the Stauffer's algorithm [14]. Each pixel in the image is modelled as a Gaussian mixture (GMM), which is updated using a spatio-temporally adapted learning rate [12]. The GMM is then used to decide if the pixel belongs to the foreground by adaptively thresholding the accumulated sorted weight of the Gaussians. Therefore, the foreground mask is formed and is then cleaned-up by shadow removal and a morphological clean-up. The weighted density of foreground pixels gives the crowd density a continuous metadata stream. Using this stream, the low-level events related to crowd appearance or disappearance are easily obtained via thresholding. An example is illustrated in Figure 3, which shows the crowd density metric as a function of time. This has three large peaks, for which the corresponding processed frames indicating the foreground blobs are given. Another frame is an example of minor activity. The extracted low level metadata indicating crowded intervals are also shown as labelled horizontal lines. A sample video demonstrating how the crowd analysis works is available online.[4]

4. SEARCH ENGINE LAYER

The search engine layer is discussed in more detail in this section, where we identify its aims and the main challenges of developing this layer.

4.1 Infrastructure Aims

The SMART search layer indexes in real-time streams of updates from edge nodes and social networks. It is built using the Terrier[5] open source search engine [11] with enhanced real-time indexing and a scalable distributed architecture to handle the large amount of streams. The SMART search layer offers an interface to services and end users to retrieve 'interesting' events and associated relevant posts in the social networks for a given query. While an interesting event is a subjective notion that likely depends on the application, the search layer can make inferences on interestingness, based on how unusual an event is, and learning from training examples of interesting events. In other words, the search engine layer uses the short-term event detection that is performed in the edge node, e.g. the low-level events detected by the crowd analysis algorithms discussed in Section 3.2, to perform unusual event detection across multiple streams of metadata. The search layer should be capable of anticipating user queries depending on their context, e.g. their location or the time of day. The search layer also offers a web service API (RESTful API) to issue queries and view results as real-time mashups aggregated from the various types of processed streams.

4.2 Challenges and Research Aims

One of the main challenges for building the search layer is the *efficient and scalable indexing* of continuous metadata streams. The search layer is built using the open source Storm[6] framework which provides a distributed processing paradigm, similar to MapReduce, that can handle streams of data in real-time. We use this architecture to distribute the workload of indexing the streams using Terrier across multiple processing nodes in a cluster. The index is distributed across various shards and an accumulator keeps track of the global index statistics. Moreover, Terrier has been enhanced to use real-time, in-memory indices, such that as soon as an update from the edge node is received, it is indexed, and made available for search. A demonstrator that uses this infrastructure to search Twitter in real-time is available online.[7]

Another main challenge in the search layer is developing an event retrieval model that can rank 'interesting' events based on a long-term pattern identification of metadata streams. The event retrieval model can make inferences on interestingness, based on how unusual an event is by comparing metadata features, such as the crowd level, observed at a specific location in a specific time to global features observed in similar areas at similar times. For example, a crowded square in a city on a Friday evening is less interesting than a crowded narrow street on a Sunday morning as this will be reflected in the background statistics of the model. Moreover, learning-to-rank retrieval approaches [9] that use features from the sensor metadata (e.g. the crowd level locally and globally) are applied to facilitate the ranking of all events happening in different locations. In addition to features extracted from the sensor metadata streams, textual evidence from the social networks can be associated to events. The event retrieval model can associate keywords to those events. For example, people who tweet about live music in a city's main square may mention the band's name or the song that is being performed.

5. APPLICATION LAYER

The uppermost layer of the SMART platform (see Figure 1) contains the software applications that can deliver the real benefits of the framework to end users. 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 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

[4] http://www.youtube.com/watch?v=akkyRu68rqE
[5] http://terrier.org
[6] https://github.com/nathanmarz/storm/
[7] http://www.smartfp7.eu/content/twitter-indexing-demo

to display for example newly-breaking events as real-time balloon pop-ups on a map.

6. OPEN SOURCE VISION

SMART is designed as an open source framework, extensible in terms of sensors, multimedia processing components and event retrieval models. As described in Section 4, the main components of the SMART search engine are built upon the existing Terrier open source information retrieval platform, allowing for the real-time indexing and retrieval of multiple and massive-scale sensor and social networks streams. The SMART open source framework is designed to benefit from the power of the open source development philosophy, by enabling application developers and organisations to build new tailored services and products on top of the SMART open source infrastructure.

In particular, SMART will form an open source community for sustaining and evolving its components. It adopts a crowd-sourcing approach to the deployment of physical sensors, social networking feeds and associated repositories, which will become searchable through SMART. Thanks to an open specification for describing data streams, the open source framework facilitates prospective information providers (including sensor infrastructure providers) to connect and contribute edge nodes and data feeds (as described in Section 3) to the SMART search engine. Hence, SMART is designed to integrate a variety of community-based sensor feeds contributed by third parties such as smart cities, sensor deployers and individuals. For example, algorithms that analyse the level of water pollution can be implemented for the corresponding sensors and made available for the fishing industry. Likewise, the SMART open source infrastructure supports virtual sensors streams such as data feeds stemming from social networks (including filters over social networks such as Twitter). In this case, SMART allows social sensors (e.g. gender analysis or sentiment analysis filters on Twitter) to be used while developing applications for smart cities. Finally, SMART adopts the business friendly MPL 2.0 (Mozilla Public License) in order to facilitate service integrators to build custom search applications in response to specific business requirements of their customers (e.g. surveillance applications). The open source application layer makes it easier for such services to be rapidly implemented. In this way, SMART intends to support both a public crowdsourcing paradigm and a private enterprise-related one. The first release of SMART is planned for the end of 2012.

7. CONCLUSIONS

We introduced an open source unified framework that allows the real-time indexing and retrieval of sensor and social streams. The framework bridges the gap between social and sensor networks and brings them closer together. The framework is currently being developed as part of the EC co-funded project SMART. We presented the research challenges for implementing the various components of the framework. In particular, the main challenges reside in developing a uniform interface to the processing algorithms of the sensor streams, the effective real-time indexing of social and sensor metadata streams and the development of efficient and effective event retrieval models. Releasing the framework as open source software and sustaining an open source community to support it is a strategical decision for a wider spread of this new technology and a wider participation from the industrial and research communities.

Acknowledgments

Part of this work has been carried out in the scope of the EC co-funded project SMART (FP7-287583). The authors acknowledge contributions from all partners of the project.

8. REFERENCES

[1] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau. Sentiment Analysis of Twitter Data. In *Proceedings of LSM'11*, 2011.

[2] A. Artikis, M. Sergot, and G. Paliouras. Run-Time Composite Event Recognition. In *Proceedings of DEBS'12*, 2012.

[3] E. Borden. Pachube Internet of Things "Bill of Rights". *http://blog.cosm.com/2011/03/pachube-internet-of-things-bill-of.html*.

[4] J. G. Breslin, S. Decker, M. Hauswirth, G. Hynes, D. Le Phuoc, A. Passant, A. Polleres, C. Rabsch, and V. Reynolds. Integrating Social Networks and Sensor Networks. In *Proceedings of W3C-FSN'09*, 2009.

[5] 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*, 2006.

[6] D. Guinard and V. Trifa. Towards the Web of Things: Web Mashups for Embedded Devices. In *Proceedings of WWW'09*, 2009.

[7] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011.

[8] A. Jeffries. A Sensor In Every Chicken: Cisco Bets on the Internet of Things. *ReadWriteWeb*, 2009.

[9] Tie-Yan Liu. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[10] D. O'Byrne, R. Brennan, and D. O'Sullivan. Implementing the Draft W3C Semantic Sensor Network Ontology. In *Proceedings of PERCOM Workshops*, 2010.

[11] 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'06*, 2006.

[12] A. Pnevmatikakis and L. Polymenakos. Robust Estimation of Background for Fixed Cameras. In *Proceedings of CIC'06*, 2006.

[13] C. Stadler, J. Lehmann, K. Höffner, and S. Auer. LinkedGeoData: A Core for a Web of Spatial Open Data. *Semantic Web Journal*, 3(4), 2012.

[14] C. Stauffer and W. E. L. Grimson. Learning Patterns of Activity Using Real-Time Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[15] K. Taylor. Semantic Sensor Networks: The W3C SSN-XG Ontology and How to Semantically Enable Real Time Sensor Feeds. In *Proceedings of SemTech'11*, 2011.