An Architecture for Detecting Events in Real-Time using Massive Heterogeneous Data Sources

George Valkanas, Dimitrios Gunopulos
University of Athens
{gvalk,dg}@di.uoa.gr

Ioannis Boutsis, Vana Kalogeraki
Athens University of Economics and Business
{mpoutsis,vana}@aueb.gr

ABSTRACT

The wealth of information that is readily available nowadays grants researchers and practitioners the ability to develop techniques and applications that monitor and react to all sorts of circumstances: from network congestions to natural catastrophies. Therefore, it is no longer a question of whether this can be done, but how to do it in real-time, and if possible proactively. Consequently, it becomes a necessity to develop a platform that will aggregate all the necessary information and will orchestrate it in the best way possible, towards meeting these goals. A main problem that arises in such a setting is the high diversity of the incoming data, obtained from very different sources such as sensors, smart phones, GPS signals and social networks. The large volume of the incoming data is a gift that ensures high quality of the produced output, but also a curse, because higher computational resources are needed. In this paper, we present the architecture of a framework designed to gather, aggregate and process a wide range of sensory input coming from very different sources. A distinctive characteristic of our framework is the active involvement of citizens. We guide the description of how our framework meets our requirements through two indicative use cases.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models; D.4.7 [**Operating Systems**]: Organization and Design—*Distributed Systems*

General Terms

Algorithms, Experimentation

Keywords

Architecture, Event Detection & Response, Real-Time

1. INTRODUCTION

The technological advancements that have occurred during the past decade in various domains, including sensors, wireless communications, location positioning technologies and the web, allow for the collection of a wide range of data. Given the abundance of information that is available from all of these sources, researchers and practitioners are now able to develop techniques and applications that monitor and react to all sorts of conditions. For instance, sensors have been used to automatically control high performance facilities, such as power plants and hotels. Health informatics also rely on devices with sensory input, to monitor exercising or more critical data, such as heartbeats and pulse. GPS signals give rise to location-based services, whereas the GPS traces can be used for trajectory recommendations, on handheld devices [6]. Finally, social media sites, such as Twitter and Facebook, contain large amounts of user generated content. These platforms serve for users to freely express their views and interests, but also to discuss them online with their social networks.

A feature shared by all of these mediums is that access is generally readily available or is fairly easy to obtain. For example, sensory devices are very cheap, and their cost is decreasing, while their capabilities are increasing. Meanwhile, smartphones are equipped with sensory devices, including accelerometers, temperature readers and GPS. They also run applications with which users can post status updates on online social media, including the rest of the contextual information. Social media sites provide access to that data through Application Programming Interfaces (APIs) over the network. Twitter [1], for example, has attracted considerable attention from the research community, due to the large user involvement with the service, but also due to its data openness policy.

In this setting it is important to efficiently process it in real-time, and combine it in meaningful ways to better understand what is going on. At the same time, offline processing can be of tremendous help, because it can help us build historical knowledge around the monitored entities. Typically there is an interplay between online and offline processing in the event detection and response process. The typical situational awareness cycle where one observes a situation, decides on an action and acts, followed by observation of the results of the action depends on online processing for its support, where everything is happening in real time. By logging everything, we are also able to do offline processing, to improve and evaluate anew the processes that are being used. As an example, consider a recurring pattern of traf-

[1] http://www.twitter.com/

fic network congestion within a city. Offline processing can help us understand and create models of user mobility. We can subsequently use these models in the online setting to identify the congestion source and suggest alternative routes to the drivers and reduce traffic jams.

The availability of diverse and extensive information allows us to address more challenging problems, such as real-time event detection and response, disaster and crisis management, emergency reporting, and sustainability applications. In this paper we focus on real-time event discovery for crisis management and emergency reporting applications. Such applications are motivated by civil protection agencies, which are responsible for aiding civilians in these occasions, and their decisions greatly depend on the quality of the data they can access. Fast response time is also critical in their line of work, given that people's lives are at stake. Allowing such agencies to have easier, faster and accurate access to the data we previously described, should improve their performance and quality of decision making, as they would be more informed regarding what is happening. However, it is imperative to present each time the most relevant information for the task at hand, which makes the overall problem even more challenging.

The goal of this paper is to initiate work on the development of a platform which will aggregate all the necessary information and will orchestrate it appropriately. A fundamental problem that arises is the high diversity of the incoming data, obtained from very different sources and having different formats. For example, sensory data are (typically) single valued, measuring a specific feature (e.g., temperature), GPS signals are comprised of $(latitude, longitude)$ pairs, whereas social media input is mostly textual, covering anything that the users want to talk about.

In this paper we focus specifically on two data sources, namely social microblogging data (such as twitter) and cellphone GPS trajectory data. The reason we focus on those two data sources is two-fold: First, they represent the two aspects of a user's experience which are becoming increasingly interwined given the recent advances of increasingly on connectivity and very efficient availability of network resources. Second, they represent potentially massive data, enabling us to explore and expand the limitations of current techniques.

Events themselves may be defined in very different scales, making a coherent understanding difficult. For example an unusually high-traffic situation in the centre of a city can be described in one twitter text message, sent by user that is complaining for a traffic jam, or by a sequence of values in the traffic sensors that are located in the area of the traffic jam. In addition to the difference of the scales, another problem is the high volume of the collected data. Twitter alone now counts more than 200 million active users, with an approximate 340 million tweets on a daily basis [2]. The large volume of incoming data is a gift that ensures high quality of the produced output, but also a curse, because higher computational resources are needed.

2. RELATED WORK

There have been significant steps in infrastructure development over the past few years, with regards to disaster management and early warning systems. For example, Flood Watch [1] developped by the DHI Group, and the Mike modelling framework [2], are built to support decision making in case of flooding. The integration of Semantic Web technologies for flood warning was also investigated during the SemsorGrid4Env project [8]. Algorithmic techniques that try to predict the likelihood of flooding, based on bayesian classification techniques were proposed in [11]. With the exception of the last one, which does not present a fully fledged system, the rest rely entirely on input from sensory data. Despite the ongoing advancements in hardware, sensor-based approaches are limited to the capabilities of the sensors. Thermal sensors, i.e. devices that can only measure temperature, can not be used for protection from flooding, and vice-versa. Moreover, sensors are only able to monitor the area where they are deployed. For example, sensory devices deployed on the river bank are unable to monitor what is going on within the city center. The only way to solve these problems is to deploy across all areas of interest different sets of sensors, one set for each emergency we want to monitor. This solution is not only costly, but also impractical to manage and maintain.

Very similar in spirit are frameworks for fire monitoring. SCIER [3] is an indicative example that dealt with this type of emergency, and relied on sensory data, thereby facing the same inefficiencies that we previously discussed. The more recent TELEIOS project [4] provides a "Virtual Observatory Infrastructure". The project combines spatial and spatiotemporal Semantic Web technologies with satellite images. Therefore, it is more closely related to remote sensing. However, a major downside is its strong coupling with the fire monitoring application. The reason is that the input data can provide only so much detail as to identify incidents of fire. In other words, even if we adapted the TELEIOS infrastructure to allow for the identification of other types of events, the results would be far from satisfactory. Satellite images are difficult to obtain, periodicity is quite small and even then they are unsuitable for locating network congestions or even earthquakes. On the contrary, social media [14] and smartphone driven [13] techniques have been shown to be able to achieve this.

To meet the needs of real-time processing for event detection and response, we need a robust and efficient underlying infrastructure. The recent *INSIGHT* project focuses on gathering, aggregating and processing a wide range of sensory input coming from very different sources. The ultimate goal of the *INSIGHT* project is to provide a real-time computational platform which will facilitate decision making processes in the presence of mission critical tasks (e.g., disaster management). In addition to the variety of data sources, another distinctive characteristic of our framework is the active involvement of citizens, as part of the entire processing cycle, by employing crowdsourcing and active learning techniques. We use a diverse set of data sources, including sensory data, GPS data, as well as user generated data. Essentially, we reinforce the *social sensor* naming convention of social me-

[2]https://business.twitter.com/audiences-twitter

[3]http://www.scier.eu/

[4]http://www.earthobservatory.eu/

dia users, who post information regarding their surrounings, in *real time*. In that respect, each user can be seen as an individual sensor, who can submit any type of information, according to their interests. With the right incentives, the user can also become engaged to submit high quality information. It is important to note that user input is not only used as a means to identify events or emergencies, for which action should be taken, but also to shed further light and describe what the event is about. Explaining the event to decision makers is of paramount importance, especially if it is a complex one, which is affected by several parameters.

Projects that use a variety of information to address event detection, and are therefore more closely related to INSIGHT, are PRONTO [5] and WeKnowIt [6]. Compared to the former, INSIGHT is interested in providing a generic computational platform, that will facilitate the detection of events of various types, ranging from network traffic (e.g., road congestion, car accidents) to natural calamities. Moreover, event detection within INSIGHT is based on input from all sensory input, including that from *social* sensors. On the contrary, PRONTO uses feedback from the users for the descriptive enrichment of events only. We also employ active learning and crowdsourcing techniques to improve the quality of the obtained results. Regarding WeKnowIt, the objectives of that project largely differ from the ones of INSIGHT: WeKnowIt uses collected information to *understand* the impact of new technologies in reporting during disaster management, whereas we use collected information to actually *guide* the decision making process. In that respect, INSIGHT also faces technical challenges, because it needs to process and combine voluminous, heterogeneous data in real-time.

3. OUR GOAL

We aim at developing the infrastructure to mine and manage the available information coming from multiple, heterogeneous sources in real-time. Such an infrastructure finds several applications, with the most important ones being in emergency response and disaster management situations. Civil protection agencies are therefore the primary users of our infrastructure, although simple users benefit as well. Civil protection agencies base their decisions on the information that they have available, and therefore, their decisions can only be as good as the data they have. Providing them with an infrastructure that intelligently aggregates heterogeneous information, that is readily available enables them to make more informed decisions.

Succintly, our **goal** is to develop a general architecture that can target a set of applications, including **flood management**, **emergency response** and **urban transportation planning**, which have a requirement for monitoring and processing information coming from different sources in real-time. Our specific objectives are:

- To develop an *adaptive*, *scalable* and *dependable*, *real-time* infrastructure for emergency monitoring. Such applications need efficient real-time processing to deliver information on time. The scalability requirement comes from the fact that today there are large data sources, including data collected by extensive sensor networks, that we must process. Adaptivity stems from the need to combine heterogeneous sources whereas dependability is required by the critical nature of the applications that we want to support.

- To develop novel methodologies for *monitoring, processing, analyzing* and *synthesizing* massive amounts of heterogeneous data for improving our ability of coping with emergencies. In this work we focus on detecting and understanding events from different data sources. Event detecting and monitoring is a general problem in several settings, and can be particularly important in emergency management situations as it can help improve the situational awareness of those tasked with managing the response to an emergency.

- To efficiently analyse diverse data sources, to correlate events identified in different scales, and to improve the accuracy of our methods, we aim to develop and use novel *crowdsourcing* techniques, and novel mechanisms for uncertainty management.

- To ensure *reusability* and facilitate *faster adaptation* of the proposed methodology. This practically translates to making the transition from the old infrastructure to the new one as smooth as possible. A system that is difficult to use or understand, is impractical for the use-cases that we consider, since we are dealing with life-critical situations.

4. EVENT DETECTING AND MONITORING

A major goal of our work is to develop an approach for event monitoring and the appropriate mechanisms that will deal with the volume of the data, as well as the heterogeneity of the data in time and scale. We focus mainly on the following types of data sources: mobile and smartphone data (e.g., GPS signals, trajectories), and social media sites (e.g., Twitter). The variety of sources is because we acknowledge that there is no "one-size-fits-all" and different sources are better suited for different tasks.

These sources produce a continuous stream of data, which are being received by the monitoring engine. Of course, each source is received by a separate module, but the monitoring engine can be seen as the entry point of all input to our system. Due to the heterogeneity of the data, but mostly because of them being inherently noisy, we need to address *uncertainty management*. The uncertainty module is required so that it provides probabilistic estimates of the events we are encountering, enriching them with confidence values. An event for which we have very low confidence of occurrence (because the sensors are too noisy), should not signal an alert. On the other hand, an event that is very rare, but for which we have high confidence, through this module, should realise an alarm to the decision makers.

Uncertainty management is not targeting one particular data source. It applies in sensor networks, because the event we see could just be a sensor failing; such an event is insignificant for civil protection agencies. It also applies to social media [9], as users may be influenced by their social peers.

[5]http://www.ict-pronto.org/
[6]http://www.weknowit.eu/

The plethora of data sources also helps us reduce uncertainty and verify (or reject) a potential event, by increasing our confidence through multiple channels of input. Note that the uncertainty management module also has a backfeed to the input sources. We have designed the module in this way, exactly for this particular reason: to be able to query additional sources, and improve our confidence of an event's occurrence or not.

Finally, there is the actual data processing layer, which processes the incoming data as the input stream is received. Therefore, for the most part, this involves stream processing techniques, to cope with the real-time nature of the setting. The same module is used to improve the models that are being used. Consequently, the models are constantly evaluated and improved, through implicit and explicit feedback received from using the system itself.

In the following paragraphs, we give some additional details to the processing frameworks of smartphone and Twitter data.

4.1 Social Media Processing

Twitter is one of the most popular networking sites in the contemporary web. It currently has around 200 million active users who post more than 340 million tweets (short messages) on a daily basis. It relies on user generated content, of small size, that is constantly updated. Users are allowed to post short messages, no more than 140 characters, without limiting them to a particular topic. Engaged users talk with friends or broadcast their interests. Twitter also has a social component aspect, allowing users to connect with each other, thereby forming a directed graph. The diversity of its users, in terms of cultural background, spoken language, interests, etc. make it a very challenging medium.

In addition to being a social media platform, due to its inherent real-time nature of constant updating, Twitter has also been a fruitful medium for news reporting, because it provides up-to-date information of what is being discussed. Indicative examples of this intrinsic characteristic include the recent political termoil in the Middle East, the Japanese earthquakes [14], or the 2007 Southern California wildfires [16].

Despite its potential as a platform for real-time event detection, there are several challenges that we need to tackle. For instance, as events are typically tied to a location in the physical world, we need to know where the users reporting the event are situated. However, only a small fraction of the users disclose that information [17]. In fact, no more than 5-6% of the users use GPS signals to indicate their location, which has led to several techniques for location extraction [7, 10, 14] from user profiles.

Figure 1 depicts an architectural overview of the components that are necessary to identify meaningful events in Twitter. The final output will be, of course, the same as the one from the Smartphone Event Detection component.

Event detection is our primary objective, and one of the means to achieve this is emotional cues. Following cognitive and affective theories [18], our event detection mechanism makes the assumption that tweets which deviate significantly from the (aggregate) norm are the result of such events. We will not elaborate on the mechanics behind event detection, as they go well beyond the scope of this paper. Succintly, we maintain an online sample of the received data and we rely strictly on data streamed by Twitter and not on query-based solutions. Therefore, our system operates as though the stream was observed by the service itself.

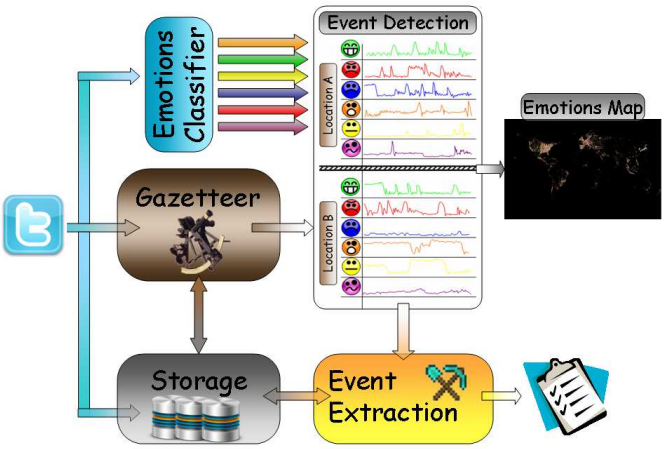

Figure 1: Architecture for Event Detection in Twitter

A prominent component of the Twitter Event Detection architecture is the Gazetteer, which performs geocoding of users, i.e., positions them on a (geographical) map. Events can also be associated with emotions, which is why we also have the Emotion Classification component. This could provide additional information to decision makers, and it could also be useful feedback for the uncertainty module component, we have discussed in the previous paragraphs. Note that our current architecture does not list a specific technique for event detection. Therefore, online clustering could be applied [5, 12], graph partinioning [19] or lookups from previously extracted events [4].

Events extracted from Twitter are immediately associated with some particular text, because the medium itself revolves around textual content generated by users. For this reason, Twitter can be used as a service to provide meaningful interpretations of events, that we encounter through other input sources (e.g., sensors). For example, after an event is identified from sensor values, we can probe Twitter users to ask them to describe the situation for us. A description of the event is very important, because it gives the authorities a better view of the magnitude of the problem. We could also enrich this information with external links, metadata, or additionally extracted information e.g., emotions. Emotions, for instance, are a valuable piece of information for the social sciences (psychology and sociology in particular), and can be used to monitor the reactions of a group of people to an event. A visualization technique that combines event summarization, associated with emotions is provided by [18].

The sheer amount of data that is produced on Twitter, also makes the problem challenging from a technical perspective. For this reason, we have built a distributed process-

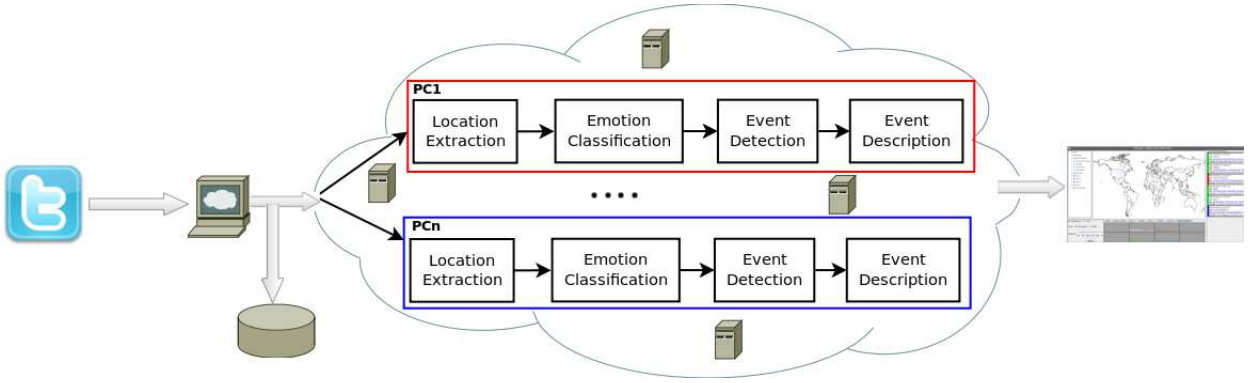

Figure 2: Processing chain of information in a distributed, online fashion

ing system, among several machines, to apply online analysis. A simplified processing chain is shown in Figure 2, where we only consider the Twitter stream as input. Note that this processing chain only considers event identification from Twitter. There are additional query-based channels, to probe Twitter for further information on a specific topic, using hashtags, tokens, etc.

4.2 Smartphone Event Processing

Smartphones and handheld devices (netbooks, iPads, etc.) are becoming more and more ubiquitous. A distinctive characteristic that these devices share, as opposed to an ordinary laptop or a desktop computer, is that users always carry them with them. Even the simplest smartphones have, nowadays, sensing capabilities like GPS, camera, microphone and accelerometer, WiFi and bluetooth communication, etc. Therefore, they can easily transmit and receive data across the network, and inform the interested parties about what is going on.

Our goal is to exploit the collective data streams generated by application software modules running on multiple user smartphone devices and shared by the users in the distributed system, to detect events of interest.

We assume that each stream of data consists of a sequence of chunks of data, called Application Data Units (ADUs); these are messages triggered locally at the user mobile phones using the sensing devices on the phones. Their exact form is application dependent, an example of an ADUs is: <user id, latitude, longitude, timestamp> (for a traffic monitoring application). The data streams may vary in size as well as volume since they may combine several types of data with different characteristics (e.g., they may contain video samples for analyzing the level of congestion in a city junction) and so we must ensure that the available system resources will be able to process the amount of ADUs generated. Smartphones are powerful enough to do some local processing rather than just sending the raw data streams, such as computing the average time to reach a destination under the current traffic conditions or triggering an alarm when there is high traffic.

Social ties of a group are important, because they are indicative of common interests. In this paper we use the concentration of groups of people in an area to hint to us that an event occurs. Therefore, when a set of users, who share social ties, are clustered in a particular area, this could be a good indication of an event. Note, however, that for the same reason, the event could be a social one (*i.e.*, attending a concert), or an emergency one (*i.e.,* fire in a building). We develop a dynamic clustering technique that allows us to cluster data streams generated from user mobile devices and then use a crowdsourcing component to ask a representative subset of the streams in the clusters to monitor. This is built on the premise that when an event occurs, users will gather around it. This finding has been validated in [13].

Assuming a data set $D = (x_1, x_2, ..., x_m)$, we aim to extract a number of nonoverlapping subsets $x_1, x_2, ..., x_n$, with $n < m$, identified as clusters, based on the clustering criterion (*i.e.,* maximum distance of any two points within the cluster should be below a *threshold*). Given a number of users in the mobile system, each described by the following information $< user\_id, latitude, longitude, timestamp >$, we use the streaming data provided by the users to cluster the users based on their geographical locations. This data is gathered by a monitoring module that runs locally on the phone, this is responsible for building and maintaining resource utilization profiles as well as transmitting the application data through the network substrate. Thus, at each time instance, the clustering component: (i) creates new data points to represent new users in the system, based on their locations, (ii) updates the locations for moving users, and (iii) removes the points that represent users that have not participated in the system for a long time period. DBSCAN [15] and Optics [3] are two of the most common density-based clustering algorithms. However, they are not designed to deal with mobile users where the clusters need to be managed and updated at runtime. The advantage of our approach is that only users that have changed their locations will need to be updated in the clusters. However, changing one point can affect multiple neighbor points in the cluster, triggering further changes to the number of clusters. We assume, that, in addition to the sensing modules and the application components run by the user, there is also a monitoring module run locally on the devices.

The crowdsourcing component is responsible to select a biased subset of users from each cluster, (k), depending on resource availability and based on the maximum amount of data units that the system can efficiently transmit and pro-

cess. The clustering component works in concern with the crowdsourcing component to achieve our goals. Clustering is applied on the GPS collected from the users, to geographically group them together in the physical world, while the crowdsourcing component performs the sampling function to determine which nodes will participate in the event detection process.

Figure 3 gives a high overview of the flow of the data from the nodes, through the applications, and the communication with the distributed stream processing system. Data is collected from the nodes. Based on the sampling policy, we select a set of nodes to participate in the clustering step, which we also apply. As new nodes are discovered, or as some are removed from the network, we reapply the clustering algorithm and subsequently sampling. This helps us identify how events evolve over time as well, and monitor them through time and space. Note that this is a distributed stream processing system, given that we have multiple participating nodes, and that we need to process this information in real-time.

5. EXPERIMENTS

In this section, we provide a brief experimental evaluation of our architecture. The first set of experiments is to demonstrate the efficiency of our system in terms of the Twitter processing chain. Given that various event detection modules could be employed, with varied efficiency values, we will focus on the Location Extraction part, for which we are using a custom solution [17].

For geocoding users, we use an RTree structure to pinpoint those who specify their location through GPS. For the rest who provide it with a textual description, we employ lookups on a custom location database (a "gazetteer"). For the purposes of this task, we built two possible gazetteers, with different information content. Both of them use the GeoNames [7] dataset as the basis. The first one, "Full", is built using the entire GeoNames dataset. The second one, "Small" uses only the fraction of the GeoNames dataset that is about inhabited places. The reason for "Small" is that several of the locations in GeoNames are about lakes, forests, etc., where users would not be found, and matches against these locations would seem most likely odd.

[7] http://www.geonames.org

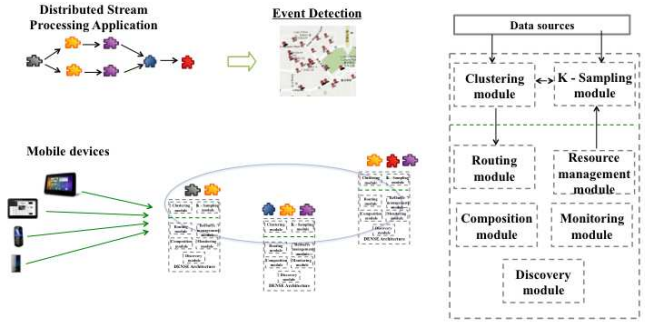

Figure 3: Event Extraction from Concetrated Groups

The experiment was to geocode – map textual descriptions of locations to $(latitude, longitude)$ pairs – 1.8 million unique locations, which have been extracted over the period of a two month crawl from Twitter. For this experiment we used a single Quad Core machine.

The size of the gazetteer impacts the amount of time needed to geocode the locations, with the "Small" dataset performing (eventually) around 10× faster than the "Full" one. As we can also see from [17], the efficiency gains are also followed by effectiveness gains as well. Note that it takes no more than 2 hours to geocode the, approximately, 2 million locations; although these numbers may sound high for real-time processing, we should keep in mind that this is a batch processed task, taking only 2 hours over a 2 month period. Note also that the system scales linearly with the number of locations that it processes. Parallelizing this task is not an issue as each location is not processed multiple times, meaning that in our distributed architecture it would take a lot less than the 2 hours. We can also employ caching mechanisms to improve the overall performance.

6. CONCLUSIONS

In this paper we presented a general architecture for real-time event discovery from heterogenous data sources, motivated by the INSIGHT project. Our long term goal is to develop a robust, distributed, real-time response infrastructure to facilitate emergency response and disaster management and aid involved agencies in such tasks. We described the type of data that we are processing in achieving our objectives, and the challenges we are faced with. Uncertainty management can be used to improve on the noisy data input from various sources, and increase our confidence in certain events, and crowdsourcing mechanisms can be used to find correlations between different datasets. We demonstrated the specific architectural decisions made for two of the major data sources we have in our system, namely smartphones and Twitter stream. We are working on the aggregation and combination of all these data sources and their derived signals, under the presence of uncertainty in the system, to produce high quality outputs, with meaningful descriptions.

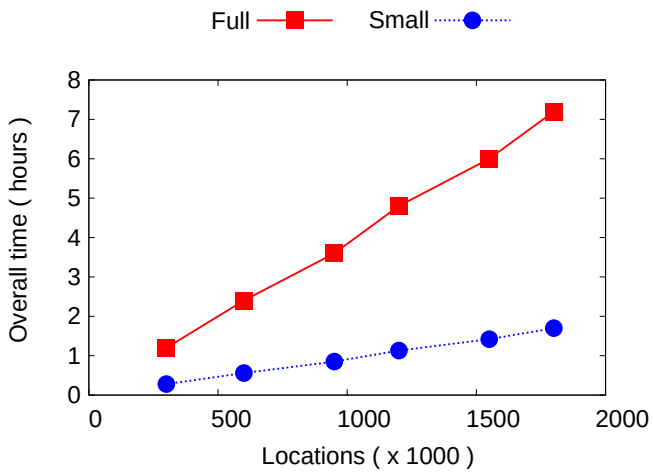

Figure 4: Efficiency of "Full" vs "Small" gazetteer

Acknowledgements

This work has been co-financed by EU and Greek National funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Programs: Heraclitus II fellowship, the EU funded project INSIGHT and the ERC IDEAS NGHCS project.

7. REFERENCES

[1] Flood watch – decision support system for real-time forecasting. Available online from: http://www.dhigroup.com/upload/publications/mike11/Skotner_MIKE_FLOOD_watch.pdf, Accessed on June 1st 2013.

[2] Mike flood. Available online from: http://mikebydhi.com/Applications/CoastAndSea/CoastalFlooding.aspx, Accessed on June 1st 2013.

[3] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: ordering points to identify the clustering structure. In *SIGMOD*, Philadelphia, PA, June 1999.

[4] H. Becker, D. Iter, M. Naaman, and L. Gravano. Identifying content for planned events across social media sites. In *WSDM*, 2012.

[5] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. WSDM, 2010.

[6] C. Costa, C. Laoudias, D. Zeinalipour-Yazti, and D. Gunopulos. Smarttrace: Finding similar trajectories in smartphone networks without disclosing the traces. In *ICDE*, pages 1288–1291, 2011.

[7] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing. A latent variable model for geographic lexical variation. In *EMNLP*, 2010.

[8] A. J. G. Gray, J. Sadler, O. Kit, K. Kyzirakos, M. Karpathiotakis, J.-P. Calbimonte, K. Page, R. García-Castro, A. Frazer, I. Galpin, A. A. A. Fernandes, N. W. Paton, O. Corcho, M. Koubarakis, D. D. Roure, K. Martinez, and A. Gómez-Pérez. A semantic sensor web for environmental decisionăsupport applications. *Sensors*, 11(9):8855–8887, 2011.

[9] M. Gupta, P. Zhao, and J. Han. Evaluating event credibility on twitter. In *SDM*, pages 153–164, 2012.

[10] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis. Discovering geographical topics in the twitter stream. WWW, 2012.

[11] M. L. V. Martina, E. Todini, and A. Libralon. A bayesian decision approach to rainfall thresholds based flood warning. *Hydrology and Earth System Sciences*, 10(3):413–426, 2006.

[12] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *SIGMOD*, 2010.

[13] I. Mpoutsis, V. Kalogeraki, and D. Gunopulos. Efficient event detection by exploiting crowds. In *The 7th ACM International Conference on Distributed Event-Based Systems (DEBS 2013)*, Arlington, Texas, USA, June-July 2013.

[14] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, 2010.

[15] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Min. Knowl. Discov.*, 2(2):169–194, June 1998.

[16] J. Sutton, L. Palen, and I. Shlovski. Back-channels on the front lines: Emerging use of social media in the 2007 southern california wildfires. 2008.

[17] G. Valkanas and D. Gunopulos. Location extraction from social networks with commodity software and online data. In *ICDM Workshops (SSTDM)*, 2012.

[18] G. Valkanas and D. Gunopulos. A ui prototype for emotion-based event detection in the live web. In *SS-KDD-HCI @ SouthCHI*, 2013.

[19] J. Weng and B.-S. Lee. Event detection in twitter. In *ICWSM*, 2011.