

SaferCity: a System for Detecting and Analyzing Incidents from Social Media

Michele Berlingerio
IBM Research
Dublin, Ireland
michele.berlingerio[at]gmail[dot]com

Francesco Calabrese
IBM Research
Dublin, Ireland
fcalabre[at]ie[dot]ibm[dot]com

Giusy Di Lorenzo
IBM Research
Dublin, Ireland
giusydil[at]ie[dot]ibm[dot]com

Xiaowen Dong
IBM Research
Dublin, Ireland
xiaodong[at]ie[dot]ibm[dot]com

Yiannis Gkoufas
IBM Research
Dublin, Ireland
yiannisg[at]ie[dot]ibm[dot]com

Dimitrios Mavroeidis
IBM Research
Dublin, Ireland
dimitrim[at]ie[dot]ibm[dot]com

Abstract—This paper presents a system to identify and characterize public safety related incidents from social media, and enrich the situational awareness that law enforcement entities have on potentially-unreported activities happening in a city. The system is based on a new spatio-temporal clustering algorithm that is able to identify and characterize relevant incidents given even a small number of social media reports. We present a web-based application exposing the features of the system, and demonstrate its usefulness in detecting, from Twitter, public safety related incidents occurred in New York City during the Occupy Wall Street protests.

I. INTRODUCTION

In this paper we present SaferCity, a system to identify and analyze public safety related incidents from social media. In contrast to the relevant public safety analytics systems, SaferCity takes into account both official crime and safety issue reports, and uses the two to build a new kind of *situational awareness* that complements the current view available to law enforcement entities. On one hand, in fact, current commercial systems used by police and other security forces rely mostly on official reports and deployed sensors such as CCTV cameras, audio sensors, emergency calls, and so on. On the other hand, in the last few years, research studies attempted to capture signals of safety issues by extracting knowledge from Social Media, a voluntary but indirect form for reporting safety issues in urban contexts.

The need for additional information beyond official crime data is illustrated by several reports that estimate the amount of unreported crime to be high. Notably, a recent report issued by the Bureau of Justice Statistics, of the U.S. department of justice states: “During the period from 2006 to 2010, 52% of all violent victimizations, or an annual average of 3,382,200 violent victimizations, were not reported to the police.” [1].

The shortcomings of official crime data are further illustrated by the research works that analyze the concept of “fear of crime” or “perception of safety”. These works commonly identify discrepancies between official crime statistics and “fear of crime” in different areas of a city. Although unreported crime can partly explain these discrepancies, researchers have

also investigated other social factors in order to explain this phenomenon.

The complexity and importance of the concept of “fear of crime” and also the documented inefficiencies of official crime data, highlight the need for a novel framework that leverages additional information resources for identifying and analyzing public safety incidents. In this project we take this challenge and develop a novel framework that employs social media content (e.g. tweets) in order to identify public safety related incidents, analyze the people’s perception of these incidents and also quantify and analyze potential spatio-temporal patterns that relate public safety incidents with urban sensed data.

The challenges of analyzing social media data are well documented in the relevant data mining literature [2]. The valuable information is usually hidden in a very large amount of noise, thus calling for advanced data analysis techniques that are able to extract information in very challenging signal/noise ratio environments. This challenge has lead most social media data mining research to focus on large scale events [3] that affect a large number of people, thus working with a more favorable signal/noise ratio. In this project, we have to address the problem of incident detection also for small events, by using data mining in a novel way, and combining urban data.

II. ARCHITECTURE

Figure 1 illustrates the modular architecture of SaferCity. We isolate our data models using an abstract layer that separates core algorithm implementations from database querying. The core components are implemented in an extensible framework enabling future expansions. At a high level, the proposed architecture supports an Application Programming Interface (API) that a user developed application can leverage to support a range of different applications on public safety in urban contexts. As User Interface layer we implemented a web application using OpenStreetMap¹ layers, that users can adopt to browse the detected events in their multidimensionality, including space, time, category, sentiment, and match with official reports.

¹<http://www.openstreetmap.org>

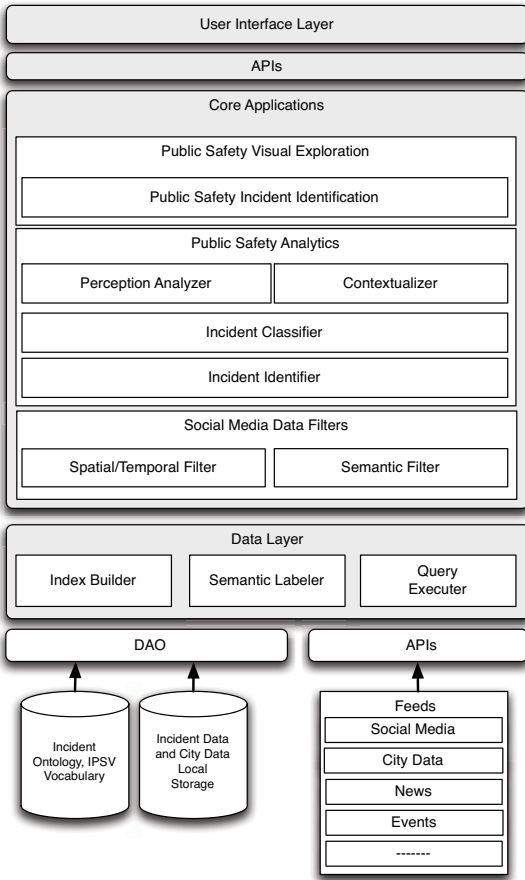


Fig. 1. High level architecture overview

The architecture is divided into the following main components: Data Layer (DL); Social Media Data Filters (SMDF); Public Safety Analytics (PSA); Public Safety Visual Exploration (PSVE); and User Interface Layer (UIL).

The following section presents the data model used in SaferCity, while sections II-B and II-C present an overview of the most important components.

A. Data Model

SaferCity was designed to work with *social data content*. An instance of such kind of data is defined by the following fields: (i) unique user identifier, (ii) location expressed in latitude and longitude, (iii) timestamp, and (iv) text. Examples of data following this model are:

- geo-located tweets collected from Twitter, using the tweets themselves as text;
- geo-located photos retrievable from various Web sites such as Flickr² or Panoramio³, using associated tags or descriptions as text.

²<http://www.flickr.com/>

³<http://www.panoramio.com/>

In addition, SaferCity uses information about public events, reported incidents and citizens complaints. Possible sources of such information are law enforcement authorities, or public data available in the Web: examples are 311 calls⁴ and crime statistics⁵. An entry in our *incident* data model consists of the following fields: (i) a unique identifier for the incident, (ii) location expressed in latitude and longitude, (iii) timestamp, and (iv) tags, keywords or a textual description associated with the incident.

In the rest of the paper, we assume to work on Twitter data, used as social data content.

B. Data Layer

The most important components of this layer are the Index Builder and the Semantic Labeller. The Query Executer is then used to query the index by using categories expanded using the output of the Semantic Labeller.

1) *Index Builder*: This component builds a Lucene⁶ index out of the incoming data. We built (and then queried) two different indexes: one for the original tweets, one for the resulting clusters (see Section II-C1). The methodology consists of the following steps. First, from each line of data (either a tweet, or a cluster), we call an analyzing module that removes stop words, performs the Porter's stemming, performs sanity check, and tokenizes the input. Then, a "document" is built from the result, and Lucene builds an index and an inverted index from all the documents. The inverted index is then used in the Query Executer, to retrieve the relevant data. At this stage, it is important to apply the same analyzing module to the query, to match the index. The Query Executer called on this index returns a ranked list of relevant documents for a given query, together with a score, that is used for post-processing filtering.

2) *Semantic Labeller*: This component provides semantic enrichment of the words contained in the social data content. Especially for data sources like Twitter, where the length of the content is limited, the information extraction task is hard, as posts are usually short, noisy, contain ungrammatical text and provide very limited context of the words they contain. We use a semantic labelling strategy to expand the meaning of each word, and use the resulting richer context for querying. The component leverages IPSV⁷, an offline vocabulary publicly available, that is used as follows. For each word in the available tweets (stop-words were eliminated) we:

- query IPSV to get the relevant corresponding labels
- consider all the broader categories in IPSV (up to level 2) with its corresponding labels
- compute a similarity score, for each word and each obtained label.

C. Public Safety Analytics

The main components here are the Event Identifier, the Classifier, and the Perception Analyzer.

⁴<http://www1.nyc.gov/apps/311/>

⁵<http://www.houstontx.gov/police/cs/stats2.htm>

⁶<http://lucene.apache.org>

⁷<http://www.esd.org.uk/standards/ipsv/2.00>

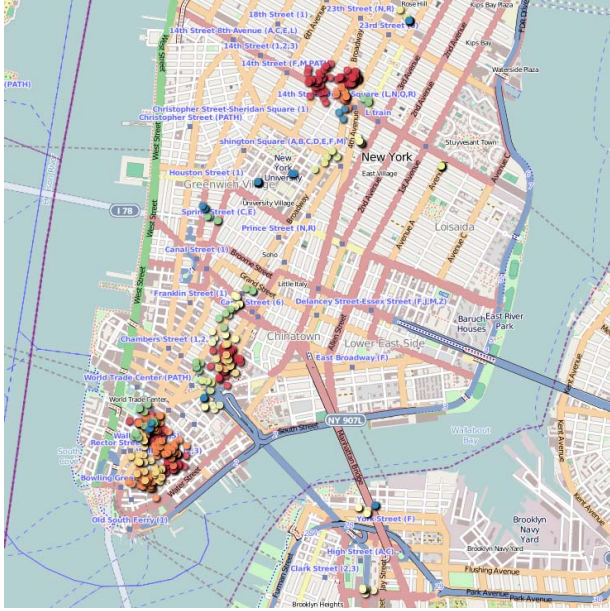


Fig. 2. Tweet Clusters for 17/11/2011. Colors represent different cluster IDs.

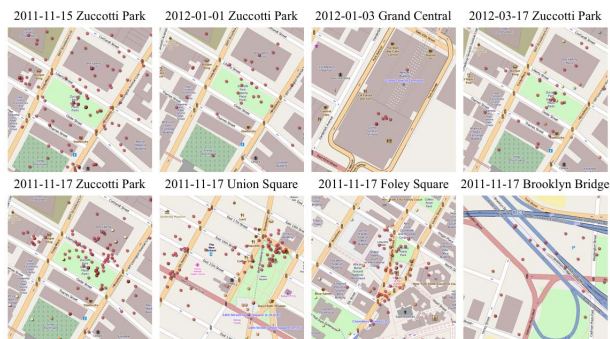


Fig. 3. Tweet Clusters and Corresponding OWS Major Events

1) *Event Identifier*: The goal of the event identifier is to detect incidents/events that take place in the area covered by our system. This component takes into account the different input resources (such as city-context information, and the Twitter stream) and is able to identify map areas where an event of interest takes place. This component also provides a short description or most relevant information regarding the identified events.

The event identification component centrally employs a spatio-temporal clustering approach that also takes into account the textual content of a tweet. The clustering method is an adaptation of the Louvain method [4], and is based on local modularity maximization.

We present an example clustering output of the proposed system, using as input the New York City geolocalized tweet stream of 17/11/2011, a date when several Occupy Wall Street (OWS) protests took place. A general overview of the clustering output is illustrated in Figure 2.

Some clusters found by the proposed system, which are related to some major events of the Occupy Wall Street protest

movement, as well as example tweets within each cluster, are shown in Fig. 3 and Table I, respectively. The major events are also listed in Table I. As we can see, the proposed clustering procedure successfully identified these major events of the OWS movement. In particular, all the four locations of protests on 17th November, 2011 mentioned by the Wikipedia timeline⁸ are identified by the event clusters we obtain.

2) *Classifier*: The purpose of this module is to assign a class to each input in the social data content. In the case of Twitter, this means adding one feature to each tweet, or each cluster, representing additional information extracted from the classifier. The additional information we could extract regards the category of the highest correlated incident. This may be one of the categories extracted by the Semantic Labeler / Filters. As input, this module takes one data record per tweet, or per cluster, containing the textual features of a tweet (such as length, whether it contains a picture, and so on), or the textual and analytical features of a cluster (text, number of tweets, users, and so on), and the score provided by querying the Lucene index for all the categories considered. When the classifier is used on a new incoming tweet, the output is in the format of an additional feature for that tweet, containing the matching category. The output will then be joined with the original tweet features (tweet id, lat, long, text, time, user id). For clusters, the classifier is able to label them with their corresponding category, for latter analysis.

3) *Perception Analyzer*: The Perception Analyzer component applies sentiment analysis to user-generated content, in this case clusters of tweets, in order to show the Twitter users' perception of the incidents. We adopt the APIs provided by Sentiment140⁹ [5] to implement the sentiment analysis. The methodology behind it consists of the following steps. First, a set of training data (tweets) are labeled as examples indicating positive or negative sentiment, based on the emoticons within the text. Second, a classifier is trained using the Maximum Entropy model, taking unigrams and bigrams contained in the text as features. Finally, the trained classifier is applied to test data to assign a sentiment score to each tweet, namely, -1 for negative, 0 for neutral, and 1 for positive sentiment. The sentiment of an incident cluster is then calculated as the average sentiment score of all the tweets within the cluster.

III. DEMO ON REAL WORLD DATA

The demo will be shown in a web browser, presenting a web application that allows the users to analyze the public safety related incidents the system has identified, and compare it with the official data such as incidents from the 311 calls. The interface of the demo is shown in Fig 4 and Fig 5. In the demo, a user can choose to display incidents detected by the SaferCity (shown as colored areas), or incidents from the 311 calls (shown as blue dots), or both. We now focus on the incident clusters detected by the proposed system. On the top of the left panel, the user first chooses a specific date to analyze the incidents. She/he will then have the choice whether to analyze all the incident clusters on the selected date, or to focus only on a specific type of incident clusters. Depending on this choice, a tag cloud will be displayed on

⁸https://en.wikipedia.org/wiki/Timeline_of_Occupy_Wall_Street.

⁹The APIs are publicly available at <http://help.sentiment140.com/api>.

Date	Event	Example tweets	Tweet ID
2011-11-15	At about 1pm, NYPD began to clear <i>Zuccotti Park</i> .	Lines of NYPD circulating inside park. Stand here, don't stand there etc. outside perimeter lined by riot police. #OWS	136593898050043905
2011-11-17	More than 30,000 demonstrated in and around <i>Zuccotti Park</i> , <i>Union Square</i> , <i>Foley Square</i> , <i>the Brooklyn Bridge</i> , and other locations through the city.	Mostly media, police right now in Zuccotti Park. We need more numbers. Get down here. #OWS #N17 Occupy wall street is occupying union square. As long as I can get home on the subway later chant on. Chant on March stretches from Brooklyn Bridge all the way back to Foley Square. Thousands lined up down Centre Street on way to bridge #n17 #OWS Crossed Brooklyn Bridge and was greeted by cop saying, Welcome to Brooklyn.	137128655636803584 137268645297532929 137315275430309888 137334153854197760
2012-01-01	New York police arrested 68 Occupy Wall Street protesters after they moved back into <i>Zuccotti Park</i> where the movement began last year.	Arrests happening now in Zuccotti Park #ows #OccupyWallSt	153361598260580353
2012-01-03	Approximately 200 Occupy protesters performed a flash mob at the main concourse of New York's <i>Grand Central Terminal</i> .	#Occupy #ows protest in Grand Central #NewYork #NYC http://t.co/Zg6k20Cr	154337396203339776
2012-03-17	Occupy Wall Street demonstrators attempted to reoccupy <i>Zuccotti Park</i> to mark the movement's six month anniversary.	Haven't seem Zuccotti like this in months. Some instigation by protestors but police seem tense today, too #M17 #OWS	181082983598530560

TABLE I. SOME MAJOR EVENTS DURING THE OWS MOVEMENT, AND EXAMPLE TWEETS WITHIN THE CLUSTERS THAT CORRESPOND TO THESE EVENTS FOUND BY THE PROPOSED SYSTEM. ORIGINAL TWEETS CAN BE RETRIEVED USING TWITTER'S APIS WITH THE TWEET ID: [HTTPS://DEV.TWITTER.COM/DOCS/API/1.1/GET/STATUSES/SHOW/:ID](https://dev.twitter.com/docs/api/1.1/get/statuses/show/:id).

the bottom of the left panel, showing the tags extracted from the tweets contained in the corresponding clusters. Once a tag cloud is generated, the user can choose to only display those clusters associated with one specific tag by clicking it in the tag cloud. The clusters will be displayed on the right panel of the interface, and will be updated according to the user's choices.

In our demo, the clusters are visualized as areas on the map that enclose the locations of all the tweets within the clusters. A user can check all the tweets in a cluster by clicking the button "show tweets". The clusters are color-coded according to the sentiment score assigned to the cluster, namely, a red cluster indicates negative sentiment about the incident, while a green cluster indicates positive sentiment. A sliding window on the bottom of the right panel allows to display clusters within specific time intervals during the day.

We present an example of analyzing a real world event using the proposed system, namely the Occupy Wall Street protest movement in New York City from late 2011 to 2012. First, as we have already seen in Fig. 3, the proposed system is able to identify major events during the OWS movement. Second, the proposed system is able to identify small-scaled clusters that are meaningful. For example, Fig. 4 shows a cluster consisting of only four tweets from three users. The corresponding tweets are shown in the top half of Table II). As we can see, the tweets within this small cluster indicate what was happening near Foley Square at that moment, what would happen soon (protesters crossing the Brooklyn Bridge), and the reaction of the police. Third, the sentiment analysis indicates people's perception of the local incidents. For instance, incidents about protests (Fig. 4) reflect negative sentiment of the crowd, while incidents about protesters cheering up (Fig. 5 and the bottom half of Table II) reflect positive sentiment.

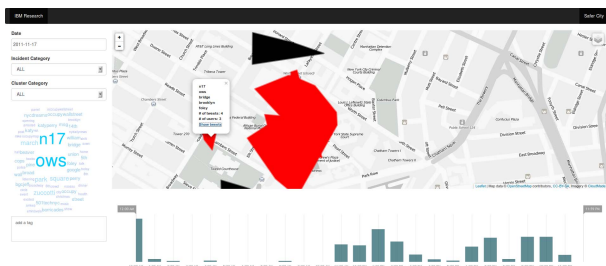


Fig. 4. A small-scaled cluster indicating the local situation of the movement, with negative sentiment of the crowd.

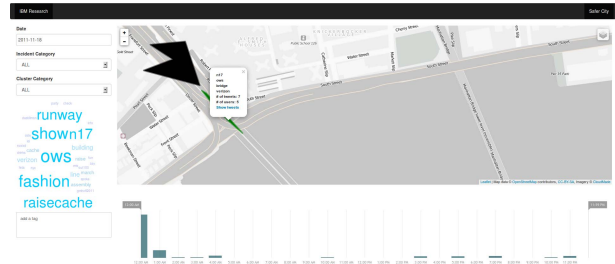


Fig. 5. Example local incidents reflecting positive sentiment of the crowd.

Date	Example tweets	Tweet ID
2011-11-17	Big numbers already at Foley Square and steady stream approaching. Safe to say thousands #n17 #OWS	137293574134956033
	Photographers get ready: everyone that will cross The Brooklyn Bridge is being given candles. #ows	137296243251625984
	12 police horses waiting on Chambers St, peaceful down here so far tho #n17 #OWS	137303344757747712
2011-11-18	Could this be any more amazing? Thanks @Verizon for the billboard. :) #n17 #ows #occupywallstreet @V1 http://t.co/ljFYE3EF	137327704461672449
	We are unstoppable! Another world is possible! #ows #N17 http://t.co/IX4orzYV	137330997086789633
	The us flag on the tower of the Brooklyn bridge seems placed just for us. Everything is possible. #ows #N17	137331368479830016

TABLE II. TWO EXAMPLE CLUSTERS FOUND BY THE PROPOSED SYSTEM.

IV. VIDEO AND REQUIREMENTS

The system can be seen in action in the video available at: <http://www.youtube.com/watch?v=d1VCZpGYvKQ>

The demo will be displayed in a Web browser, and requires a working Internet connection to display correctly. To display the demo, we will need a power socket, a table, and a video projector or a large monitor. The presenter will bring a laptop to run the system.

REFERENCES

- [1] L. Langton, M. Berzofsky, C. Krebs, and H. Smiley-McDonald, "Victimizations not reported to the police, 2006-2010," *The American Economic Review*, pp. 1-22, 2012.
- [2] R. Li, K. H. Lei, R. Khadiwala, and K. C.-C. Chang, "Tedas: A twitter-based event detection and analysis system," *International Conference on Data Engineering*, 2012.
- [3] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: Real-time event detection by social sensors," *International Conference on World Wide Web*, 2010.
- [4] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 10, p. P10008(12pp), 2008.
- [5] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *Technical Report, Stanford University*, 2009.