Separating the Wheat from the Chaff: Events Detection in Twitter Data

Andrea Ferracani
Media Integration and Communication Centre - University of Florence
Firenze, Italy
andrea.ferracani@unifi.it

Daniele Pezzatini
Department of Information and Communication Technologies, Universitat Pompeu Fabra
Barcelona, Spain
daniele.pezzatini@upf.edu

Lea Landucci
Media Integration and Communication Centre - University of Florence
Firenze, Italy
lea.landucci@unifi.it

Giuseppe Becchi
Media Integration and Communication Centre - University of Florence
Firenze, Italy
giuseppe.becchi@unifi.it

Alberto Del Bimbo
Media Integration and Communication Centre - University of Florence
Firenze, Italy
alberto.delbimbo@unifi.it

ABSTRACT

In this paper we present a system for the detection and validation of macro and micro-events in cities (e.g. concerts, business meetings, car accidents) through the analysis of geolocalized messages from Twitter. A simple but effective method is proposed for unknown event detection designed to alleviate computational issues in traditional approaches. The method is exploited by a web interface that in addition to visualizing the results of the automatic computation exposes interactive tools to inspect, validate the data and refine the processing pipeline. Researchers can exploit the web application for the rapid creation of macro and micro-events datasets of geolocalized messages currently unavailable and needed to improve supervised and unsupervised events classification on Twitter. The system has been evaluated in terms of precision.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; **Web mining**; **Web interfaces**; **Browsers**; **Users and interactive retrieval**;

KEYWORDS

Unsupervised Event Detection, Data Mining, Twitter, Visualization, Datasets creation

ACM Reference format:
Andrea Ferracani, Daniele Pezzatini, Lea Landucci, Giuseppe Becchi, and Alberto Del Bimbo. 2017. Separating the Wheat from the Chaff: Events Detection in Twitter Data. In *Proceedings of CBMI, Florence, Italy, June 19-21, 2017,* 5 pages.
DOI: 10.1145/3095713.3095728

CBMI, Florence, Italy

DOI: 10.1145/3095713.3095728

1 INTRODUCTION

Social networking sites such as Twitter have become platforms where people communicate and share knowledge on a daily basis through text messages, photos, videos. This huge amount of social data produced by participatory and crowd-generated sensing represent an opportunity to extract knowledge and give valuable insights on urban dynamics. The flow of information from Twitter is realtime, covers diverse topics and can describe actual events. These events can even be quite small such as parties, companies' presentations, road accidents and so on. Well-established networking mechanisms can improve the information gain through the analysis of this flow's dynamics. Number of messages in time, social reactions such as *likes*, *retweets* and direct replies, geolocation along with multimedia features can be processed in order to detect the occurrences of events using statistical models. In November, 2016 Twitter counted 317 active millions users. Roughly 80% of these users access the platform through a mobile device and about 2% of them choose to share their GPS location. This is a small percentage but nevertheless it is a large number of people who produce significant contextual information, especially in densely populated urban areas.

In this paper we propose a method for unknown event detection which relies on this geolocated data. The method uses a lightweight statistical approach and can alleviate common issues on this subject related to computational cost and systems scalability. Most of the works in the literature exploit content-based unsupervised approaches for event detection with a considerable computational complexity (see Sec.2). The huge amount of text streams to be processed in realtime, the sparsity of geolocalized data and the noisy nature of Twitter messages are some of the main obstacles researchers have to cope with. These issues make often unfeasible an implementation in a real system.

To address the challenge we based our method on geographic spatial grids and implemented it in a processing pipeline, explained in Sec. 3, which combines several algorithms for statistical analysis without exploiting highly computationally intensive mining techniques. The approach focuses on the analysis of temporal and

spatial characteristics of tweets distributions in order to detect abnormalities and accumulations. The method can be effective for macro and micro-events detection as it accounts for the historical time series in terms of volumes and density of data. Supervised approaches can of course perform better but they may require an hard and time consuming work for tweets discovery and labelling not feasible for unspecified events. In order to support further research on Twitter event detection, our method has been implemented in a web interface which provides users with tools to validate and categorize events automatically discovered by the system and to store them in datasets of geolocated tweets. To the best of our knowledge, there are no datasets of Twitter macro and micro-events available to researchers which contain exclusively geolocated data and cover diverse topics. Sec. 5 reports the results of an evaluation of our system in building a dataset of events with geolocated messages published in London and New York.

2 RELATED WORK

Previous studies have addressed the problem of detecting events in social data and specifically on Twitter through the identification of abnormalities in its temporal flow. Exploited features are mainly frequency and density of terms, hashtags, named entities, reactions, emoticons. All these works use a variety of techniques ranging from K-means clustering, SVM, gradient boosted decision trees to generative language models and temporal query expansion [1, 5, 6, 8, 10]. Wang et al. [14] improve clustering quality enriching Twitter messages with term expansion on *Wordnet*. Generative models and statistical clustering on textual data from Twitter has been used more recently in [13] and [4]. Nguyen et al. [7] consider keyword occurrences (Occurence-Score) over time, number of participants involved (Diffusion-Degree) and speed of information spread (Diffusion-Sensitivity) to calculate the probability that an event occurred on the basis of term score distribution. Hierarchical clustering of terms and *Wordnet* expansion have been used in [12] for emerging event detection. The authors observe changes in events' popularity defined as number of messages in events clusters. Irregularities in the rate of messages are exploited in [9] and in [3] where also location and topical clustering is performed. As regard to applications, closest to our work is the *CityBeat* [15] system which detects abnormal signals combining time series and classification methods based on spatial, meta, textual and historical features. One of the main issues of existing approaches on event detection from tweets is the computational cost of extracting and elaborating a lot of features. Twitter messages are composed by very small sentences filled with mispelled words, hashtags, symbols, urls that need to be cleaned and normalized. This noisy realtime data is huge and not easily manageable from a computational point of view. Studies such as [4, 7, 13, 14] aim to function in real-time on all this data but their performance is poor in terms of processing time and system scalability. Furthermore performances are evaluated only on small *corpora* or not evaluated at all.

For addressing these challenges, we propose a processing pipeline designed keeping in mind that an event occurs in time and space. From our perspective, in order to reduce processing time and implement algorithms in a usable tool it is feasible to reverse common approaches and to take into account exclusively geolocated data. This may be regarded as a limit but we think it is definitely a *plus* for the main objective of our system which is the implementation of a method for the detection of geolocalized macro and micro-events exploited by a tool for events datasets creation. In fact, this choice 1) decreases false positives; 2) filters the information reducing the amount of data to be processed; 3) allows the rapid creation of datasets of events through our tool, that otherwise would require a considerable amount of work for searching and inspecting the Twitter knowledge-base.

Our pipeline combines some algorithms and techniques, explained in Sec. 3, and is implemented in a web system. The system provides an exploratory interface which allows to refine and improve the detection adjusting the spatial and temporal parameters exploited by the algorithms, allowing a fine-grained analysis which works on subsets of data. The main outcome of the work is the design and implementation of a lightweight and configurable system for *1)* the semi-automatic detection of macro and micro-events of urban geo-data extracted from Twitter (see Sec. 3) and *2)* the easily creation and management of datasets of micro-events, currently unavailable for research studies on Twitter data[1].

3 DETECTION AND MINING

The proposed processing pipeline used in our system contemplates three main steps: *a)* tweets extraction; *b)* abnormalities detection; *c)* mining and visualization.

Tweets extraction. tweets are extracted daily through the Twitter API using the Java library twitter4j[2] and stored in a mySQL database for post-processing.

Abnormalities detection. we use a grid-based method to analyze all the geo-referenced data in a certain urban area. The area is divided arbitrarily in cells of predefined dimensions. Tweets in each cell of the grid are analyzed with our method which combines temporal and spatial density analysis. The idea of the method is to firstly divide tweets spatially into a grid, then to use time series analysis (exploiting DTW and Crest Detection) to detect anomalies in the volume of tweets of the cell, and to finally exploit spatial clustering to infer possible events from anomalies. For each cell c, we create a time series V containing the number of unique users per hour who published geolocated tweets in the cell c. We define a time interval T (e.g. 24 hours), and we divide the initial time series V in windows of size T obtaining a set of time series V_i, with $i \in [0, N_{windows}]$. We then compute the average time series $\overline{V}$. On this basis we perform our method for abnormalities and event detection which consists in a pipeline of three core algorithms described below:

(1) **DTW** Dynamic Time Warping is an algorithm which allows to measure the similarity and the distance between two time series. DTW is widely adopted in information retrieval to cope with deformations of time-dependent data [11]. For each time window i, we compute the DTW distance between the time series V_i and the average time series $\overline{V}$, obtaining a measure of distance d_i. To detect windows of time where the distribution of tweets is unusual, we

[1] video available at https://vimeo.com/miccunifi/twitter-events-detection
[2] http://twitter4j.org/en

consider the average of the distance $\overline{d}$ and the standard deviation σ_d. All the time series for which $d_i > \overline{d} + \sigma_d$ or $d_i < \overline{d} - \sigma_d$ are considered as abnormalities and sent to the step 2 (Crest Detection).
(2) **Crest-detection** is an algorithm which uses peaks windowing in order to detect anomalies in data distribution. We consider the time series marked as abnormal in step 1. For each time-step t, $t \in [0, T]$, we compare the values of unique tweetters in that hour $v(t)$ with the values of the time series in a temporal interval of size Γ_t that precedes and follows t. A peak is detected at time t if $v(t) > \sum_{j=1,..,\Gamma_t} v(t-j) + \epsilon$ and $v(t) > \sum_{j=1,..,\Gamma_t} v(t+j) + \epsilon$. Values of Γ_t and ϵ are set to 2 as default (customizable through the interface, see Sec. 4)
(3) **DBSCAN** Density-based spatial clustering of Applications with Noise [2] is a data clustering algorithm that given a set of point in the space groups together the points that are closer in the distribution. The algorithm divides the points (i.e. tweets with *lat, lng*) in 'core points', 'border points' and 'noise points'. A point p is a 'core point' if at least a minimum number $minPts$ of points are comprised in a distance ϵ and are directly reachable from p. Mutually density-connected 'core points' form a cluster. A point q is a'border point' part of the cluster if a path exists between p and q so that all the 'core points' in the cluster are density-reachable from any point of the cluster itself. All the other not reachable points are considered 'noise points'. An event is proposed by the system if at least one cluster is detected by DBSCAN for the tweets posted during the temporal interval of the peaks detected in step 2. As default the system sets $\epsilon = 5$ meters and $minPts = 3$ (customizable using the interface).

Mining and visualization. Once the event clusters has been identified by the DBSCAN algorithm, the event is visualized on the web interface and positioned in its center of mass with respect to the geolocalization of each tweet (see Sec. 4). Text mining is performed on the tweets that are part of the cluster in order to show content features of the detected event. We extract: *I)* most frequent words; *II)* hashtags; *III)* named entities (i.e. timestamps, names of persons and organizations, places); *IV)* attached photos and videos; *V)* part-of-speech tagging. Visualization of metadata and related multimedia material is essential in order to help users in verifying the correctness of the event detection and to build categorized datasets.

4 THE WEB INTERFACE

The Web Interface has been developed in Java and deployed as a servlet in a Tomcat container[3]. The interface has been designed with the main goal to visualise on a map the results of the automatic event detection pipeline described in Sec. 3. Furthermore, it exposes semi-automatic tools which may help researchers to tune the exploited algorithms in order to improve the event detection. Not least, the system allows the creation of datasets of events from Twitter.

The interface provides two main views (see Fig. 1), both map-based [4], with two different access levels: *1)* the visualization view shows on the map the events automatically detected by the system; *2)* the validation view allows an authorized user to confirm (or not) the correctness of the detection and to customize the search.

[3]http://tomcat.apache.org/
[4]Interactive maps are provided using http://leafletjs.com/

In the visualization view each event is represented by a marker that can have three color: blue, red and green. Blue pins are the events detected by the algorithm but still not validated by a human; red and green pins instead are events respectively misclassified or correctly classified as confirmed by a user. The user can search and zoom the map and define a temporal range for the visualization of events detected by the algorithm with the defaults parameters. Each marker can be activated in order to open an info-box window which shows the data and metadata associated with the event: the hour, the category and all the extracted features (tweets, word occurrences, related multimedia, named entities, POS tagging). Authorized users can access the editing and validation view. The view shows a transparent grid super-imposed on the geographic area of interest. The dimensions of the cells of the grid are predefined

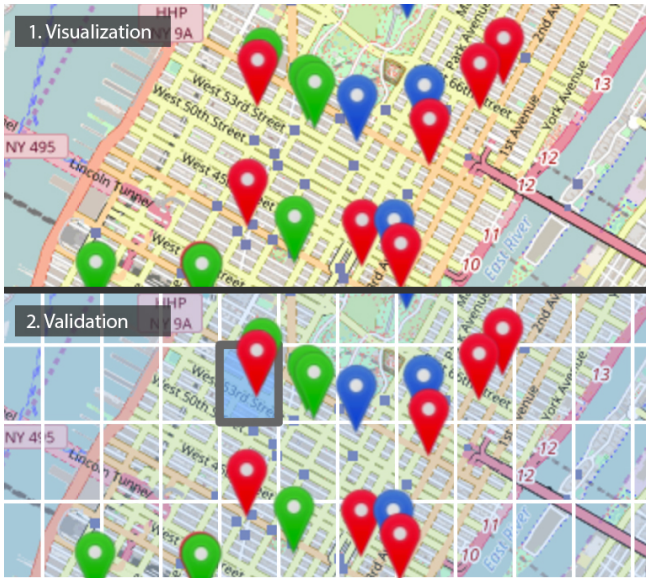

Figure 1: The two main views of the system: 1. Visualization. 2. Editing and Validation

and the grid is positioned arbitrarily to cover all the geolocalized tweets published in a configurable radius. The user can select a cell, as shown in Fig. 1, in order to show an info-box where all the events detected in that area can be inspected. In this modal interface the user is also provided with advanced graphical widgets through which he can adjust the several parameters used by the algorithms. Hence, the computation can be started again asynchronously in order to discover events previously not detected by the algorithm with the default parameters. Configurable parameters are: *1)* the position and the dimensions of the grid cell (Fig 2.1); *2)* the time interval of the event detection; *3)* the time period and/or periodicity over which to calculate the average and the standard deviation of the DTW distances for the cell; *4)* the threshold value over the average *plus* the standard deviation beyond which the system reports an abnormality; *5)* the time window and the ϵ used in the crest-detection algorithm (Fig. 3); *6)* the $minPts$ and the ϵ used by the DBSCAN algorithm for the identification of clusters of tweets on the peaks detected by the Crest Detection algorithm (see

Fig. 2.2). The flexibility provided by in tuning parameters as well as the configurability of spatial cells dimensions, density and time intervals allow the system to work in realtime on subsets of data with good performance. Furthermore, the choice between periodicity and/or temporal continuity for the computation of historical data can improve the search. In general big events tend to be periodical (e.g. football games are usually played every two weeks) and an abnormality could be detected comparing, for example, the time series of the day to the time series of the last ten days. On the contrary, a micro-event such as a small company meeting can increase the number of tweets in a certain day with respect to a periodicity average (e.g. every Thursday) but not in a continuous time interval.

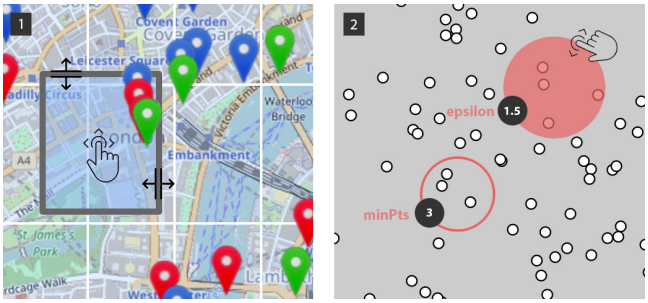

Figure 2: In 2.1: the draggable and resizable widget to select a specific area on the map. In 2.2: ϵ and *minPts* of DBSCAN can be defined dragging the circles' circumference on a distribution of points.

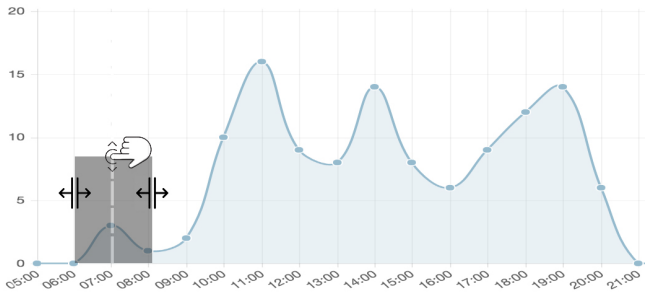

Figure 3: Time series of Twitter users who published geolocalized posts in Trafalgar Square, London, on 2016, April 1. The system detected 4 events at 7am, 11am, 2pm and 7 pm. ϵ and time interval can be changed interactively resizing the gray rectangle on the time series plot.

5 EVALUATION

To evaluate the detection accuracy of our system we manually validated the events automatically discovered between March 31 and April 9, 2016. The tweets published in the city centers of two big cities in these ten days, London and New York, were analyzed. This time period was chosen since the interval registered the highest number of tweets in the year. In details 17176 users published 44932 tweets in London, while 17378 users published 43186 tweets in New York provided with geolocalization. The system detected a total of 1240 events, 340 in London and 900 in New York using the algorithms with default parameters. The significant difference in the events' count is probably due to the diversity in population density in the two city centers (30000/km^2 in Manhattan and between 10.000/km^2 and 15.000/km^2 in the central districts of London). The overall error rate of the classification was 0.43 with 190 confirmed events in London and 516 in New York. This is a very good result in terms of precision, not far from state-of-the-art supervised approaches for event detection which range from 0.64 to 0.85 [3]. Results of a recent method following a more similar approach on geolocated tweets and Instagram photos [9] achieved a precision of just 0.20. The recall of the system has not been computed due to the unavailability of Twitter annotated datasets of macro and micro-events provided with geolocated messages.

6 CONCLUSION

In this paper we present a lightweight method for the automatic detection of unknown macro and micro-events exploiting geolocalized data from Twitter. The system uses a combination of algorithms to discover possible events using a pure statistical approach. The method is exploited by a web system which helps researchers in building datasets of geolocalized events. Default parameters of the algorithms can be changed on the fly in order to refine the detection and to validate and categorize the events proposed. These data can be useful to other researchers for improving supervised and unsupervised event detection and classification techniques on Twitter messages.

ACKNOWLEDGMENT

Research partially supported by MIUR Cluster project Social Museum and Smart Tourism.

REFERENCES

[1] Hila Becker, Mor Naaman, and Luis Gravano. 2011. Beyond Trending Topics: Real-World Event Identification on Twitter. *ICWSM* 11, 2011 (2011), 438–441.

[2] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.

[3] Tao Ge, Lei Cui, Baobao Chang, Zhifang Sui, and Ming Zhou. 2016. Event detection with burst information network. COLING.

[4] Jey Han Lau, Nigel Collier, and Timothy Baldwin. 2012. On-line Trend Analysis with Topic Models:\# twitter Trends Detection Topic Model Online.. In *COLING*. 1519–1534.

[5] Kamran Massoudi, Manos Tsagkias, Maarten De Rijke, and Wouter Weerkamp. 2011. Incorporating query expansion and quality indicators in searching microblog posts. In *European Conference on Information Retrieval*. Springer, 362–367.

[6] Donald Metzler, Congxing Cai, and Eduard Hovy. 2012. Structured event retrieval over microblog archives. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 646–655.

[7] Duc T Nguyen and Jai E Jung. 2017. Real-time event detection for online behavioral analysis of big social data. *Future Generation Computer Systems* 66 (2017), 137–145.

[8] Ana-Maria Popescu, Marco Pennacchiotti, and Deepa Paranjpe. 2011. Extracting events and event descriptions from twitter. In *Proceedings of the 20th international conference companion on World wide web*. ACM, 105–106.

[9] Søren B Ranneries, Mads E Kalør, Sofie Aa Nielsen, Lukas N Dalgaard, Lasse D Christensen, and Nattiya Kanhabua. 2016. Wisdom of the local crowd: detecting local events using social media data. In *Proceedings of the 8th ACM Conference on Web Science*. ACM, 352–354.

[10] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*. ACM, 851–860.

[11] Pavel Senin. 2008. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* 855 (2008), 1–23.

[12] Sayan Unankard, Xue Li, and Mohamed A Sharaf. 2015. Emerging event detection in social networks with location sensitivity. *World Wide Web* 18, 5 (2015), 1393–1417.

[13] Maximilian Walther and Michael Kaisser. 2013. Geo-spatial event detection in the twitter stream. In *European Conference on Information Retrieval*. Springer, 356–367.

[14] Jun Wang, Yiming Zhou, Lin Li, Biyun Hu, and Xia Hu. 2009. Improving short text clustering performance with keyword expansion. In *The Sixth International Symposium on Neural Networks (ISNN 2009)*. Springer, 291–298.

[15] Chaolun Xia, Raz Schwartz, Ke Xie, Adam Krebs, Andrew Langdon, Jeremy Ting, and Mor Naaman. 2014. CityBeat: real-time social media visualization of hyper-local city data. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 167–170.