Event Detection from Social Media Data

George Valkanas [1], Dimitrios Gunopulos [2]
Department of Informatics & Telecommunications, University of Athens
Athens, Greece
{[1]gvalk, [2]dg}@di.uoa.gr

Abstract

Microblogging platforms, such as Twitter, Tumblr etc., have been established as key components in the contemporary Web ecosystem. Users constantly post snippets of information regarding their actions, interests or perception of their surroundings, which is why they have been attributed the term Live Web. Nevertheless, research on such platforms has been quite limited when it comes to identifying events, but is rapidly gaining ground. Event identification is a key step to news reporting, proactive or reactive crisis management at multiple scales, efficient resource allocation, etc. In this paper, we focus on the problem of automatically identifying events as they occur, in such a user-driven, fast paced and voluminous setting. We propose a novel and natural way to address the issue using notions from emotional theories, combined with spatiotemporal information and employ online event detection mechanisms to solve it at large scale in a distributed fashion. We present a modular framework that incorporates these ideas and allows monitoring of the Twitter stream in real time.

1 Introduction

The web ecosystem has changed dramatically over the last decade, with the users becoming its driving force. A major shift has been that users are no longer passive observers but actively engage in online activities and experiences. Social media sites, such as Facebook, Twitter, Flickr, etc, have been at the forefront of this change, providing the necessary platforms for users to share aspects of their everyday lives online. For instance, Twitter now counts more than 200 million active users, with an approximate 400 million "tweets" on a daily basis [1]. Users can post short messages, up to 140 characters, mimicking a web-based version of the cell-phone SMS technology. The result is a constant flow of user generated content, arriving at varying rates depending on various factors, and is usually referred to as the Twitter stream.

Social media are complementary to online blogs (web logs), where the former contain snippets of more up-to-date information, while the latter are used for expressing ones thoughts, ideas, beliefs and are the result of a more thought-through process. The speedy nature of social media sites has earned them the name "*Live web*" or "*Now web*". In that respect, these platforms may serve as real-time news reporting and / or crisis-management services, as exemplified with the recent political termoil in the Middle East, with Japanese earthquakes [1], or the 2007 Southern California wildfires [2]. Given their prominent role in

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

[1] https://business.twitter.com/audiences-twitter, access Aug 2013

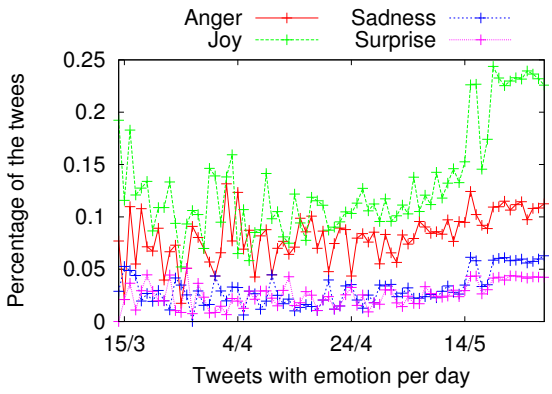

Figure 1: A timeseries of daily emotions from Twitter, between March 15 and May 24 2012

disseminating information today, it comes as no surprise that social media sites and their properties have come under considerable attention by both the academia and the industry.

One of the basic applications for analysing social media, is the problem of identifying real-life events as they happen or short after from their impact in social media. Generally we take *event* to mean an important phenomenon with a local extend and a temporal dimension in the physical world. Despite the obvious advantages in being able to do so, automatically identifying real-life events from social media data is not easy. Some of the challenges are: i) The large adoption means that we must process in *real time* voluminous amounts of data. ii) The content is usually *short*, noisy, and diverse in terms of location, languages and topics. Finally, iii) user location is also a scarce commodity leading to several techniques for location extraction [1, 3–5].

Taking into account these impediments, it is no surprise that most existing works that deal with event detection in Twitter simplify the problem by focusing on detecting events of specific event type, monitoring the stream for specific terms, or #hashtags (i.e., user generated topic labels). Clearly such approaches are useful but limited to work only when the event can be described by a small set of terms, e.g., âĂIJ[..] now shaking [..]âĂİ for earthquakes. Detecting new events by such means is difficult as the descriptive terms have to be known a priori.

Motivated by these shortcomings of existing work, we address the problem of detecting events in a stream of short-form messages, focusing on Twitter. The main goal is to devise techniques that work regardless of the category the events belong to. We take a novel approach and employ techniques grounded on influential theories of emotions, such as *Cognitive* and *Affective* [6]. According to these theories, users feel a need to express themselves as a result of an event.

Our goal is to use the Twitter stream to access such reactions. Moreover, we argue that such tweets will not be a flat description of the event, but will also convey the user's *emotional state*, partially disclosing how it affected them. An event can then be modeled as a *time-* and *place-* related phenomenon, which triggered a significant change in the emotional state of a (potentially large) group of people and our goal is to automatically capture such sudden changes. Figure 1 validates our claim: We plot the relative occurrence of the 4 most prominent emotions, from a sample of the Twitter stream, between May and March 2012. We ommit neutral tweets, which we assert to be non-informative. Surges in *anger* in early April are related with the Syrian uprising, whereas the high values of *joy* towards the end are due to the Champions League final, and the Eurovision song contest.

The rest of the paper is organized as follows: Section 3.1 discusses our event detection model and algorithmic approach, followed by Section 3 which describes the system that realises our approach. Section 2 presents related work on the event detection problem. Section 4 concludes our work and presents future directions.

2 Event Modeling And Detection

We begin by formalising the event detection problem. Following [7], *An event* $\mathbf{e}$ *is a real-world phenomenon, that occurred at some specific time* $\mathbf{t}$ *and is usually tied to a location* $\mathbf{l}$. However, using social media data, we can mainly monitor the aftermath of the event, i.e., its effects on actual people and how these are reflected in people's reactions in social media. According to influential theories of emotions [6], events will impact the users that experienced it, who will be urged to externalize their reactions, e.g., tweet about it. We expect such spontaneous reactions to convey a user's emotional state, i.e., *how* the event affected them. Making this motivation more concrete, we state our problem as follows:

Problem Statement 1: [Event Detection] Given a time ordered stream of tweets as input, identify those messages which i) alter significantly and abruptly the emotional state of a (potentially) large group of users, and ii) can be traced back to event e.

This definition fits well with an outlier detection formalisation, whereby we observe a sudden and significant change in the emotions of users, with respect to the recent history, as a result of an event taking place. However, in our definition we do not monitor individual users, using aggregate counts instead. Monitoring the reactions of individual users is very inefficient in terms of resources; however more a important problem is the ethical questions raised regarding a user's privacy as well.

To address these limitations, we use *aggregate* information from large, geographically associated groups user. Users are clustered together according to their geographical location, which we extract from available information. We then monitor the emotional state of each geographically distributed group, independently of the others and report an event when the group's *cumulative* emotional state changes suddenly. Note that this approach covers inherently the part of the definition that wants the event to affect large groups of users.

Instead of putting all users to a single group, which has no local coherency, we decompose $\mathcal{G}$ into smaller groups $\mathcal{G}_i$ and organize them hierarchically. We denote $\mathcal{G}_i^j$ as group i at level j, assuming leaf nodes at $j = 0$. The hierarchy can be administrative (e.g., country, state, etc.), or constructed algorithmically, e.g., via hierarhical clustering. For a fixed level j in the hierarchy, it holds that $\cup\mathcal{G}_i^j = \mathcal{G}$ and $\cap\mathcal{G}_i^j = \emptyset$, and $\mathcal{G}_i^j = \cup\mathcal{G}_k^{j-1}$. This decomposition offers a trade-off of high-level granularity versus a higher need in resources.

Each group $\mathcal{G}_i^0$ is then monitored by a virtual sensor s_i. Each s_i processes all of the tweets from that group. Upon arrival, each tweet is classified to one of 7 emotions: the 6 basic emotions suggested by Paul Ekman [8]: *anger, fear, disgust, happiness, sadness, surprise*, plus a *none* state. Tweets of the *none* state are not considered further, on the grounds that they are uninteresting, e.g., they reflect a mundane task. Sensors aggregate the rest of the incoming tweets along the temporal dimension, for each emotion separately. Using an *aggregation interval* a (e.g., a=1min), each s_i produces a single value for each emotion, which is the respective *count* of tweets for that emotion during a. The aggregation interval acts as a discretization unit, to cope with the streaming nature of the medium. The sensor operates over the w most recent points with a sliding window. The combination of a and w define the history that the sensor keeps track of.

Example: Assume, for instance, a sensor s_i, with a = 5 minutes and w = 12. The sensor maintains a history of the past 5 $\times$ 12 = 60 minutes. Every 5 minutes, s_i will process a single value for each emotion, extracted from the tweets received during that interval from the group of users that it monitors. The oldest point will be discarded and the new one will take its place.

2.1 Approximating the Emotional State Distribution

Given that a user's emotional state is a result of several factors, it would be unfounded to assume that it will follow a predefined distribution, much less a static one. To approximate it efficiently in an online fashion we estimate the Probability Density Function (*PDF*) of the emotional distribution of each group $\mathcal{G}_i$, and we do that through kernel estimators. According to kernel estimation, each point distributes its weight in around it,

and the *kernel function* $k(x)$ describes how this is done. The distribution we want to approximate is given by $f(x)$

$$f(x) = \frac{1}{|\mathcal{T}|} \sum_{r \in \mathcal{R}} k(\overline{r} - \overline{x})$$

Here, $\mathcal{T}$ is the actual set of values that we want to approximate, $\mathcal{R}$ is a data sample maintained online by each s_i, and $k(x)$ is the kernel function. We opt for the Epanechnikov kernel function, which has a closed form integral, and can thus be computed very efficiently, given by:

$$k(x) = \begin{cases} (\frac{3}{4})^d \frac{1}{B_1 B_2 .. B_d} \prod_{1 \le i \le d} (1 - (\frac{x_i}{B_i})^2) \\ \qquad\qquad \text{if } \forall i, 1 \le i \le d, |\frac{x_i}{B_i}| < 1 \\ \\ 0, otherwise \end{cases}$$

where B_i is the kernel's bandwidth, computed with Scott's rule [9], $B_i = \sqrt{5}\sigma_i |\mathcal{R}|^{-\frac{1}{d+4}}$, and σ_i is the standard deviation for the i-th dimension (i.e., emotion). For simplicity, we ignore the interplay of emotions, and set $d = 1$. Although values need to be normalized in the $[0, 1]^d$ space, we do not find this really restrictive: A straightforward solution is to normalize with the maximum value allowed by the system's architecture (e.g., $2^{32} - 1$ for int). Alternatively, system specification requirements can indicate the load it must sustain, which will also be an upper bound (within constant factor) on the values it can process.

To approximate the data distribution, we need i) a random sample over the data that fall within the window w, and ii) the standard deviation σ of the values in w, both of which can be easily maintained online. We use "chain sampling" [10] to produce the random sample. Chain sampling selects a point s from the sample to evict and replaces it with the new point p, regardless of s being expired or not. Sampling and smoothing using a kernel function can be seen as an indirect way for filtering out spurious bursts while improving the scalability of the system.

2.2 Event Detection

We can now use the kernel density estimator to identify changes in the data distribution. The rationale is to identify events on the basis that the most recent aggregate emotional state observed by sensor s_i was not "as expected", according to s_i's history. Therefore, if a sudden change was observed, this could be caused by an external phenomenon. To characterize a new point p as a significant deviation, we first compute its probability mass over the sample $\mathcal{R}$, by evaluating the quantity

$$P(p, r) = \frac{1}{|\mathcal{R}|} \int_{[p-r, p+r]} \sum_{t_i \in \mathcal{R}} k(x - t_i) dx$$

The value r defines a neighborhood range, within which to search for points from $\mathcal{R}$. From the definition of the Epanechnikov kernel, the values need to be in the $(p_i - r - B_i, p_i + r + B_i)$ range, to contribute to the integral. If $P(p, r)$ is below a certain threshold, we say that p is an outlier, i.e. a significant change was detected in the emotional state of the observed population. Since this could be the result of an occurring event, we should trigger additional mechanisms to describe it. Therefore, event detection is decoupled from event description.

3 The TwInsight System

From the description of an event e and our event detection mechanism, it should be clear by now that we need the following information: a location l, the time of occurrence t, a set of keywords to describe it, and the emotions that were elicited as a result of the event. Figure 2 shows a schematic view of the components required to extract that information and their interaction[2].

[2]Storage image by Barry Mieny, under CC BY-NC-SA license.

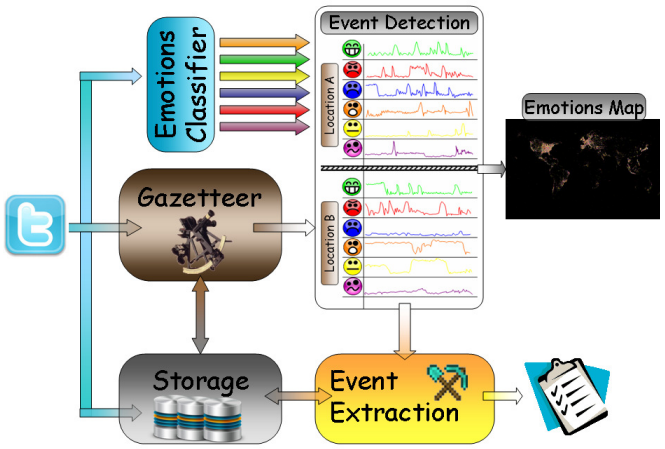

Figure 2: Schematic interaction of our system's components

The Twitter stream is our system's input, feeding two components, namely the *emotions classifier* and the *location extraction* subsystem. Locations are extracted through a custom built component [5], whereby each incoming tweet is mapped to a location. Tweets are then forwarded to the virtual sensor s_i responsible for the location it was mapped to.

Meanwhile, the tweet has been classified to one of the 7 emotions that we use. Neutral tweets are not considered for further processing, but are stored nonetheless. It is worth noting that this approach allows for an elegant integration with spam detection mechanisms: spam tweets can be cast to the *neutral* class, thereby preventing them from any subsequent processing.

When a sensor receives a tweet for further processing, we know its location and which of the 6 basic emotions it has been cast to. For each emotion, separately from the rest, the sensor counts how many tweets it has received during the last aggregation interval a. Each count is the input to a separate instance of our event detection mechanism, one for each emotion. Therefore, on each sensor, there are 6 instances of event detection mechanisms, executing simultaneously. Each event detection module updates its values and identifies whether a surge, i.e., an event, in any emotional state has occurred.

If an emotional surge was identified (i.e., an event), we report the end time of the aggregation interval as the event's time of occurrence t. Additionally, the tweet ids that caused the peak for that emotion are passed to the event extraction mechanism, which is responsible for summarizing the event. This step will provide the descriptive keywords of the event, and its operation is subject to the detection of an event. Since event detection and description are decoupled, several techniques can be used to describe the event: term frequency or TF-IDF score, summaries, etc. In any case, the user will be presented all of the necessary information: *location, timestamp, emotion* and *description*.

Table 7: Average Component Processing Time (ms)

	Location Extraction	**Classification**	**Event Detection**	**Total**
TIME:	3.36	0.35	0.001	3.72

Table 7 illustrates the efficiency of our system, **TwInsight** [11], where we show the average time taken by each component to apply its functionality on a newly received tuple. Table 8 provides some examples of events extracted by applying our method to a stream of tweets obtained between April and May 2012. A contextual user interface can also facilitate the presentation of this information, as described in [12].

Event 1 is related to the goal by Bayern's football player, Thomas Müller, in the Champions League

Table 8: Sample Summary of 15 Prominent Events Identified By *TwInsight*

ID	Emotion	Where	When (GMT)	Description
1	Joy	Germany	19/05, 20:23	thomas bayern championsleague cfc mueller muller müller
2	Joy	UK	19/05, 20:29	didier drogba f... beauty enjoying fair gal gaz goal great
3	Sadness	Canada	20/05, 22:42	died b... breaking mio robin singer @rodneyedwards gib gibb opa
4	Anger	Canada	20/5, 15:19	@ctvcalgary aime ambition chacun earthquake femme frais http://t.co/0hJEez9Q italy kills
5	Anger	US	20/5, 11:23	@Mou2amara alive assad onus prove regime shawkat showusshaukat syria

(CL) 2012 final, that took place on May 19. The goal was scored in the 83rd minute of the match, i.e. on 22:23 CEST (20:23 GMT). This places our finding the event the moment that it actually occurred and was posted. We identify similar tweets in Canada and Spain, at the *exact* same timestamp. Clearly, the event is related with Joy.

Event 2 is about the equilizer goal by Didier Drogba in the CL finals. The goal was scored in the 88th minute, i.e. on 22:28 CEST (20:28 GMT), and we identify several joyous tweets on 20:29, right after the goal.

Event 3 is about the death of Bee Gee's singer Robin Gibb. He was pronounced dead at 23:30 BST (22:30 GMT) on May 20th [3], and a surge in sad tweets is seen at 22:42, only 10' after his death.

Event 4 is about the earthquake in Italy, on May 20, that resulted in the death of six people.

Event 5 refers to Assef Shawkat, deputy Minister of Defense of Syria. On May 20, 2012, there was a claim he had been murdered [4], and tweets requesting proof were posted. We have also found tweets on 26th and 27th of May regarding the Houla Massacre of the Syrian civic war which occurred on May 25. We ommit such tweets, as they contain URLs to pictures of immense brutality.

From the list of events presented above, there are two things we would like to point out:

- Our approach is able to identify events of various types. We see events related to sports, earthquakes, popular personalities, and politics.
- We are able to identify such events promptly, as indicated by the first three events. This means that such a method is not only useful as a news reporting tool, but could be crucial in dealing with emergency and disaster management situations.

4 Related Work

Event identification from Twitter is gaining attention. Early works focus on events of specific types, e.g. earthquakes [1] or news [13]. The idea is to whitelist specific keywords and phrases, but such approaches are destined to fail when the event type is not known in advance. The technique we present here was introduced in [14].

A closely related concept is *trending topics*, i.e., terms which gain in popularity over a period of time. However, trends are not necessarily indicative of events; rather the contrary, since they are *always* present.

[3] http://www.bbc.co.uk/news/entertainment-arts-18140862

[4] http://newsfromsyria.com/2012/05/20/asef-shawkat-assassinated/

They are also prone to topical groups, e.g., a big fan base, and could be the result of recurring phenomena, such as TV shows, or *memes*, e.g., the "Follow Friday" (#FF) hashtag.

Type-independent techniques typically employ online clustering methods. Though such methods may be appropriate for some slow-paced settings [15], the data volume makes them unfit for microblogs [16], They are also sensitive to popular terms or large groups of users with similar interests, and can be easily gamed by spammers [17]. Finally, as shown in [11], these techniques require extremely careful data cleaning and preprocessing to be able to work in practice.

Taking into account these impediments, early techniques simplified the problem by focusing on a specific event type [1,18,19]. They then monitor online data for specific terms that can be used to describe the event. However, this can only work when the event can be described by a handful of terms, e.g., "[..] *now shaking* [..]" for earthquakes. Clearly such approaches cannot detect new events for which the descriptive terms are unknown a priori. Recent approaches also correlate information coming from other sources with data from the twitter stream to understand events [20].

Online clustering solutions [15,16,21] are used to identify events and trends in Twitter. Such approaches generally suffer from scalability issues [16, 22], and coping with the increasing volume of the data is a research issue itself.

5 Conclusion

In this paper, we focused on the problem of automatically identifying events from the *Live Web as they occur*. We combined notions from emotional theories with spatiotemporal information, and tackled the problem using online event detection techniques. We integrated our ideas in a modular framework and experimentally demonstrated the validity and scalability of the method.

In future work we will work to develop the system along several thrusts: i) improve the performance of location extraction method, by applying online location clustering, using GPS signals, and by using information about the Twitter graph to estimate the location of a tweet from the location of related Twitter users, ii) improve the event description by incorporating novel summarization techniques, iii) improve the classification accuracy to filter uninformative points.

Acknowledgements: The authors would like to thank the data annotators. 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, THALIS - GeomComp, THALIS - DISFER, ARISTEIA - MMD" and the EU funded project INSIGHT[5].

References

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

[2] 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.

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

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

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

[6] M. Mikolajczak, V. Tran, C. Brotheridge, and J. J. Gross, *Using an emotion regulation framework to predict the outcomes of emotional labour*. Bingley, UK: Emerald, 2009.

[5] www.insight-ict.eu

[7] H. Becker, F. Chen, D. Iter, M. Naaman, and L. Gravano, "Automatic identification and presentation of twitter content for planned events," in *ICWSM*, 2011.
[8] P. Ekman, W. Friesen, and P. Ellsworth, *Emotion in the human face: guide-lines for research and an integration of findings*. Pergamon Press, 1972.
[9] D. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*, ser. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1992.
[10] B. Babcock, M. Datar, and R. Motwani, "Sampling from a moving window over streaming data," in *SODA*, 2002.
[11] G. Valkanas and D. Gunopulos, "How the live web feels about events," in *CIKM*, 2013 (to appear).
[12] ——, "A ui prototype for emotion-based event detection in the live web," in *CHI-KDD*, 2013, pp. 89–100.
[13] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling, "Twitterstand: news in tweets," in *SIGSPATIAL-GIS*, 2009.
[14] G. Valkanas and D. Gunopulos, "How the live web feels about events," in *ACM CIKM 2013*, 2013.
[15] H. Becker, M. Naaman, and L. Gravano, "Learning similarity metrics for event identification in social media," ser. WSDM, 2010.
[16] F. Alvanaki, S. Michel, K. Ramamritham, and G. Weikum, "See what's enblogue: real-time emergent topic identification in social media," in *EDBT*, 2012.
[17] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@spam: the underground on 140 characters or less," in *CCS*, 2010.
[18] E. Benson, A. Haghighi, and R. Barzilay, "Event discovery in social media feeds," in *ACL-HLT*, 2011.
[19] D. A. Shamma, L. Kennedy, and E. F. Churchill, "Tweet the debates: understanding community annotation of uncollected sources," in *WSM*, 2009.
[20] E. M. Daly, F. Lecue, and V. Bicer, "Westland row why so slow?: fusing social media and linked data sources for understanding real-time traffic conditions."
[21] M. Mathioudakis and N. Koudas, "Twittermonitor: trend detection over the twitter stream," in *SIGMOD*, 2010.
[22] T. Lappas, M. R. Vieira, D. Gunopulos, and V. J. Tsotras, "On the spatiotemporal burstiness of terms," *PVLDB*, vol. 5, no. 9, 2012.