How the Live Web Feels About Events

George Valkanas
Dept. of Informatics & Telecom.
University of Athens
gvalk@di.uoa.gr

Dimitrios Gunopulos
Dept. of Informatics & Telecom.
University of Athens
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 all of our key ideas and experimentally validate its superiority, in terms of both efficiency and effectiveness, over the state-of-the-art using real life data from the Twitter stream. We also present empirical evidence on the importance of spatiotemporal information in event detection for this setting.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

Keywords

live web; event identification; sentiment analysis; emotions

1. INTRODUCTION

Microblogging services, such as Twitter, Tumblr and Plurk, are rapidly gaining in attention, by all involved parties: academia, industry, and the real content providers: the users. 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.

Although blogs are also user generated, the significant advantage of microblogs lies in their *constant* updating, which 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 [35], or the 2007 Southern California wildfires [40]. This characteristic of fast change, along with the imposition of an upper length limit, is what sets microblogs apart from other social networking services.

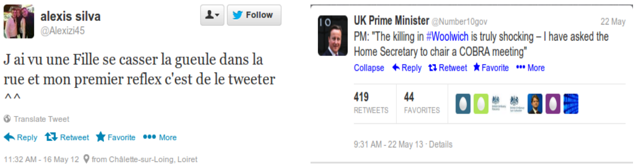
Unfortunately, automatically identifying real-life events from microblogs is not that easy. By *event*, we mean important phenomena with a local and temporal dimension in the physical world. Some of the challenges are: i) The large adoption means that we must process in *real time* voluminous amounts of data. ii) The content (text) is *short*, very noisy, with a lot of slang and personal style, and diverse in numerous ways, regarding location, languages and themes. Finally, iii) the precise location of a user is generally scarce leading to several techniques for location extraction [11, 18, 35, 41].

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 a specific event type [8, 35, 38]. They then monitor the stream for specific terms, or *#hashtags* (i.e., user generated topic labels). However, this can only work when the event can be described by a handful of terms, e.g., "[..] *now shaking* [..]" for earthquakes. Clearly, it is impossible to detect *genuine* events by such means, as the descriptive terms are unknown a priori.

Another alternative is to use online clustering [2, 7, 27] or term burstiness [20, 22], so that *trends* emerge as a set of frequently co-occurring terms. However, these techniques suffer from scalability issues, as they are known to be quite inefficient even for a small fraction of the Twitter stream [2, 22]. They are also sensitive to popular terms or large groups of users with similar interests. Spammers are also known to use such terms in their tweets, in order to "blend in" and obtain a higher visibility for their posts[14], only making matters worse. Combined with highly personal [33], and poor writing style the terms describing the event may take a while to surface with these approaches. Furthermore, a trend is *not* necessarily indicative of an event. Rather the contrary, since they are *always* present, as users constantly discuss their interests and popular terms emerge. They can also be the result of recurring phenomena, such as a prominent hashtag, e.g. "Follow Friday" (#FF), or misleading at times: "Dear santa" and "Merry Christmas" were trending at some point in May and June 2012, respectively, despite the fact that they are really out of season. In fact, as we experimen-

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

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.
Copyright 2013 ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505572.



alexis silva @Alexizi45

J ai vu une Fille se casser la gueule dans la rue et mon premier reflex c'est de le tweeter ^^

11:32 AM - 16 May 12 from Châlette-sur-Loing, Loiret

UK Prime Minister @Number10gov 22 May

PM: "The killing in #Woolwich is truly shocking – I have asked the Home Secretary to chair a COBRA meeting"

419 RETWEETS 44 FAVORITES

9:31 AM - 22 May 13 · Details

(a) Everyday incident (b) Woolwich incident

Figure 1: Tweets reflecting real-life events. Permalinks:
a) http://twitter.com/137650823/status/202677925663866880
b) http://twitter.com/Number10Gov/status/337244366181634050

tally show, clustering approaches are also ineffective, when trying to return events in a timelier manner, in such a noisy setting.

Objective: In this paper, we address the problem of detecting events in a stream of microblogs. The main challenge is to devise techniques that work regardless of the category such events belong to, e.g., sports, politics, natural phenomena, etc. To achieve this, we employ techniques grounded on influential theories of emotions, such as *Cognitive* and *Affective* [28]. According to these theories, users are urged to express themselves due to an event from the real world. Figure 1 shows indicative tweets, based on real life events of varied gravity. In both cases the users externalized their thoughts as a result of a real life event, but, as the user in Figure 1a) puts it, tweeting about that event was his *very first* reaction.

By monitoring the Twitter stream we can access these 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 2 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, the Eurovision song contest etc.

To address the sheer volume of data, we employ online event detection techniques applied on user input, thereby reinforcing the *social sensors* naming convention. More specifically, we use *aggregate* information, making our approach scalable and efficient.

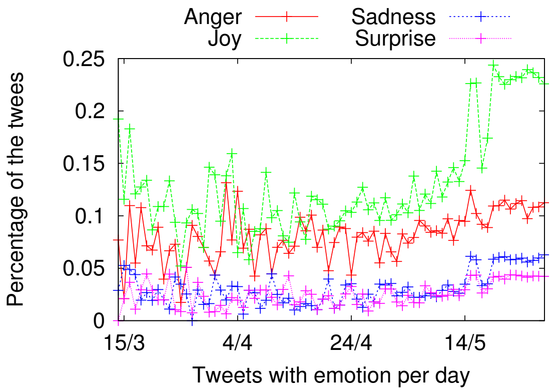


Figure 2: A timeseries on the daily emotions identified in the Twitter stream, between March 15 and May 24 2012

The general idea is to group users together and assign them to a virtual sensor, which will monitor their emotional state over time and space, in an online fashion. Information aggregation also ensures that the identified events are of interest to a large group of people. Our technique can be run in parallel, achieving high throughput and scalability, managing to efficiently process large volumes of data.

Overall, our contributions can be briefly described as follows:

- We set up a rigorous framework to aggregate emotions as a means of identifying real-life events.
- We evaluate the importance of emotions, coupled with temporal and spatial information in event detection, and demonstrate their vital role in this type of data analysis.
- We present a detailed end-to-end architecture of our approach, which we compare against the state-of-the-art in event detection, using a large crawl of Twitter data, received in a streaming fashion.

We also note that, contrary to alternatives, our approach inherently serves a dual purpose: detect new events **and** monitor a group's emotional reaction to it. This could prove extremely useful in decision making or social sciences.

The rest of the paper is organized as follows: Section 2 presents existing work on related material with our approach. Section 3 discusses our event detection model and algorithmic approach, followed by Section 4 which gives an engineering perspective of our system. Section 5 contains the experimental evaluation, and Section 6 concludes our work and presents future directions.

2. RELATED WORK

Despite our paper's ties with psychology and sentiment analysis, it is impractical to provide a detailed overview of these fields. Therefore, we focus on key aspects that relate to our problem and discuss research regarding event detection from microblogs, which is our main objective.

Psychology: Emotions are a major discipline in psychology, and several theories have been proposed to understand them [24] from various aspects, including evolutionary, social and cultural, as well as procedural. The basic intuition of our work can be seen to have grounds on procedural as well as cognitive and affective theories [28]. The latter, in particular, argue that emotions are the result of an external stimulus (i.e., an event), which will influence a person's attitude or behavior. Under such an affective state of mind, users are more likely to externalize their thoughts.

Sentiment analysis: Sentiment analysis and opinion mining [32] have been broadly studied in various domains and settings. Textual content is classified to a positive or negative class, and works have been proposed for Twitter in particular [4, 13]. Extensions deal with strength of polarity or by considering more target classes [9, 21, 23, 29]. Our current research differs from this body of work in that sentiment analysis is for us a tool to achieve our goal, i.e., event detection of real world phenomena. Therefore any such proposed technique can fit our algorithmic framework.

Event detection in Social Media: Event identification from Twitter appears to receive an increasing interest lately. Early works focus on events of specific types, e.g. earthquakes [35] or news [36]. They whitelist specific keywords and phrases, e.g. "[..]*now shaking*[..]", or sources of information, respectively. Evidently, these approaches are inapplicable for type-independent event detection, in a medium as dynamic and diverse as Twitter.

A closely related concept is *trending topics* or *trends*, i.e., terms which gain in popularity over a period of time. Trends are practically bursty phenomena [20] of term or hashtag cooccurrences. However, trends are not necessarily indicative of events; rather the contrary, since they are *always* present. For instance, a big fan base discussing their popular music idol, easily results in a trending topic, regardless of an event actually happening. They are also related with recurring phenomena, such as TV shows, or *memes*, e.g., the "Follow Friday" (#FF) hashtag. Given that our definition of an event ties it to a specific location in the physical world, techniques for spatiotemporal burstiness [22] might be considered. However, these share the same inefficiencies with the classical bursty approaches, and are also computationally expensive.

A type-independent approach was proposed in [7], where the authors applied online clustering through appropriate similarity measures. However, their methodology was meant for Flickr, an online photo-sharing service, with characteristics very different from Twitter: i) Shooting a photo requires the physical presence of user *u* at location *l* at time *t*. This information is not always available. ii) Users select the tags of their photos carefully, in order to maximize their visibility. iii) Microblogs are voluminous and updated at a very fast pace.

The most closely related work to ours is [44], which is the current state-of-the-art for event detection in Twitter. The authors employ wavelet-based techniques to capture important differences in the "*energy*" of individual terms in sliding windows. Tokens are then used to represent nodes in a graph, where edge weights encode strength of cooccurrence between terms. Subsequently, they apply a modularity-based graph partitioning algorithm to obtain groups of terms that share similar burstiness patterns. Evidently, this method has huge memory requirements, as it must maintain a sliding window with the occurrences of every token, even if it was encountered only once, because it may become bursty in the future. To address the quadratic complexity of the clustering step, the authors filter out tokens with the *median absolute deviation*. However, when applied online, this measure filters out important tokens as well. Moreover, at the conceptual level, the authors' claim that "emotional expressions are not useful in defining events". We argue the exact opposite, motivated by emotional theories. We return to this approach in our experimental section.

The works in [6, 34] aim to better understand the temporal information in tweets (e.g., "[..]*tomorrow I*[..]"), and build full calendars of events. Finally, we note the works in [16, 31] that deal with the quality of extracted events from Twitter. These techniques address an orthogonal problem, and they could be applied to our output.

3. MODELING & DETECTING EVENTS

In this section we formalize our definition of an event, and proceed with our problem statement. Similarly to [5], we define an event as follows:

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, we are only able to monitor the aftermath of the event, i.e., its effects on actual people who provide their input. According to influential theories of emotions [28], the event will have a significant impact on the emotional state of the users that experienced it. Because of this, they will be urged to externalize their emotional state, i.e., the way they feel, and will be inclined to post a message about it. Therefore, we can model the emotional state of a user as the number of tweets they post conveying one of several emotions or moods [23]: *excited, sad, angry, confused*, etc. Since an event from the real world is by definition tied to a location, we expect the *first responders* to be geographically linked as well.

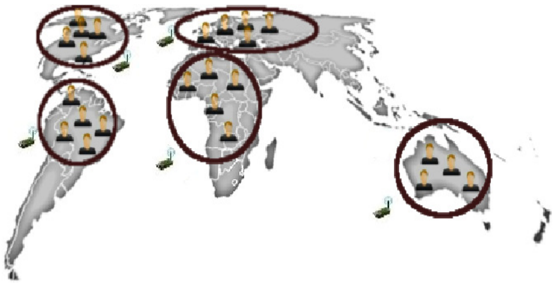


Figure 3: An example assignment of groups to virtual sensors.

Taking all that into account, our problem can be stated as:

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 nicely with an outlier detection formalisation, whereby we observe a sudden and significant change in the emotions of users, with respect to the recent history. Monitoring individual users is very inefficient resource-wise, and will not provide significant clues regarding the event anyhow. It also raises ethical questions at best, as being very intrusive on a user's privacy.

To overcome these limitations, we use *aggregate* information, extracted from larger user groups, $\mathcal{G}_i$. Users are clustered together according to their geographical location, extracted 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 that leaf nodes are 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 $\mathcal{G}_i^j = \cup\mathcal{G}_k^{j-1}$. Evidently, this decomposition is a trade-off, providing high-level granularity versus a higher need in resources.

We then assign each group $\mathcal{G}_i^0$ to a virtual sensor s_i, which *senses* (i.e., reads) all of the tweets from that group. Sensors at higher levels gather information from their children. Figure 3 shows an example of user grouping, with their assigned virtual sensor nearby. [2]. Upon arrival, each tweet is classified to one of the emotions that we monitor. Using an *aggregation interval* a (e.g., a=1min), the sensor produces a single value for each emotion, which is the respective *count* of tweets conveying that emotion during that period. 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. This results in a much simpler model than the more intricate, 2-stage, multi-level wavelet coefficients of [44]. The combination of a and w specify the history length, based on which the sensor will identify events.

Assume, for instance, a sensor s_i, *with a = 5 minutes and* w = *12. The sensor maintains a history of the past 5 × 12 = 60 minutes.*

[2] World Map image from http://www.wpclipart.com/

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.

3.1 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. In fact, we need to approximate it and maintain it efficiently in an online fashion. To achieve this, we can estimate the Probability Density Function (*PDF*) of the distribution, by observing the reactions of each group $\mathcal{G}_i$. Non-parametric models are a great fit for this purpose and kernel estimators have been shown to achieve good performance for this task [15], while being efficient.

Kernel estimation is based on the idea that each point distributes its weight in its surrounding area, and the *kernel function* describes how this is done. The function $f(x)$ which describes the distribution to approximate is given by the following equation

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

Here, $\mathcal{T}$ is the actual set of values that we want to approximate, $\mathcal{R}$ is a sample of the data, that each sensor s_i maintains, and $k(x)$ is the kernel function that describes how each data point distributes its weight. Given that the choice of the kernel function has little significance over the estimation output [37], we use the Epanechnikov kernel, which has a closed form integral, and can thus be computed very efficiently. The Epanechnikov kernel is given by the following equation

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

where B_i is the kernel's bandwidth, computed with Scott's rule [37], $B_i = \sqrt{5}\sigma_i|\mathcal{R}|^{-\frac{1}{d+4}}$. The kernel is suitable for multi-dimensional data and σ_i is the standard deviation for the i-th dimension (i.e., emotion), which can be efficiently and accurately maintained in an online fashion. For simplicity, we ignore the interplay of emotions, and set d = 1. Finally, we note that values are normalized in the $[0, 1]^d$ space. However, we do not find this really restrictive: As a straightforward approach, we can normalize with the maximum value allowed by the system's architecture (e.g., $2^{32}-1$ for int). Alternatively, we could rely on system specification requirements regarding the load it must sustain, which will also be an upper bound (within constant factor) on the values it can process.

Since we operate under a sliding window model, we need to efficiently approximate the distribution of the data which currently fall within the window. Figure 4 graphically portrays this requirement, demonstrating for two consecutive time instances the contents of a sliding window (points in blue) and their respective PDF. As time advances (from top to bottom), new points arrive and expired ones are evicted. Therefore, we must update our kernel estimation at each timepoint; using data aggregation during time intervals, instead of monitoring a stream, makes this computation tractable.

In order to approximate the data distribution, we need to i) maintain online a random sample over the data that fall within the window w, and ii) keep track of the standard deviation σ of the values within w. Both of these values are very easy to maintain in a streaming environment. We use "chain sampling" [3] to produce the random sample, which will randomly select a point s from the sample to evict, regardless of s being expired or not, and replace it with the new point p. Although we could maintain the entire stream, given enough resources, this is not necessarily a good idea, as shown in Section 5. Moreover, sampling can be seen as an indirect way for filtering spurious bursts.

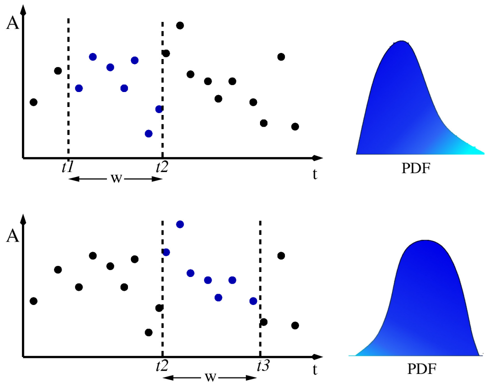


Figure 4: Approximating the data distribution in a sliding window.

3.2 Event Detection

Having our online kernel density estimator in place, we can now use it to identify changes in the data distribution. The rationale is to identify events on the basis that the most recent aggregate emotional state of users was not "as expected", according to what we have seen so far. Therefore, if a sudden change was observed, this could be caused by an external phenomenon.

Due to our "chain-sampling" approach, we always maintain a sample which reflects the latest distribution from the data, and consequently, the most recent emotional state among the users. Similar problems have been examined in sensor networks [39], but we have the advantage that i) we can maintain the full window, if we want to, as we are not as heavily constrained on our resources, and ii) an event of significance will have a more lasting effect on the users, so we expect at least one point to be inserted in the sample.

To characterize the new point as a significant deviation, we first compute its probability mass over the sample $\mathcal{R}$, according to our kernel $k(x)$. More specifically, for each new point p we evaluate 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 is the 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 that probability $P(p, r)$ is below a certain threshold, we say that this tuple is an outlier. In our setting this means that 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.

4. THE TWINSIGHT SYSTEM

In this section we discuss the components that make up our system, as well as the data and its workflow. There are several components, in order to provide the desired functionality, each of which works independently of the others. Consequently, we can substitute any one of them with better alternatives.

Twitter Data: Our only source of information is the Twitter stream, receiving tweets through the service's streaming API [3]. Having access to the gardenhose we receive a 10% sample of all public tweets that are posted to the service. Unlike previous techniques, we do not rely on external sources [31, 25] or whitelists [36].

Emotion Classification: Emotions are a key ingredient in our approach, and constitute the basic input to our algorithm (following the tweet). We use a set of 6 emotions proposed by American psychologist Paul Ekman [12]: *anger, fear, disgust, happiness, sadness, surprise*. We also use a neutral (*None*) state, to describe the absence of an emotion. We do not consider neutral tweets, as we think that they are uninformative.

In order to extract emotions from tweets, we have trained a classifier using both structural and semantic features. We consider general punctuation, such as exclamation marks, question marks, quotation marks, etc. We also consider letter capitalization / duplication (e.g. "*yesssss!!!*" instead of "*yes*"), number of retweets, whether it is a retweet (RT) itself, number of mentioned entities, hashtags and urls. The basic rationale behind using structural features of the tweet is that we expect truly spontaneous reactions to contain these characteristics. Moreover, these features are very easy to extract, because they are either directly provided by Twitter (e.g., retweets), or can be extracted through shallow parsing.

We also compiled lists of emoticons from online resources (e.g., Wikipedia) and by manual inspection of the tweets, which are good indicators of sentiment [13, 4]. Finally, we use lists of words and phrases which are known to be related to emotions. For this purpose, we use the Affective WordNet [26] vocabulary, and the moods dataset [23] which contains several sentiment-tagged words.

It is important to note that classifying tweets to emotions is harder than classic sentiment analysis. The reason is not only the additonal target classes, but rather the inherent ambiguity of certain words and the way we use them. For instance, according to Affective WordNet, "amazing" is related to *surprise*. However, in the sentence "That book was amazing!", the intent is to express *joy*. Sentiment analysis would correctly identify the sentence as "positive", but an emotions classifier may easily mislabel it. Similarly, *disgust* can easily be misinterpreted as *anger*, as in "I hate eggplants!!!!", but in either case the sentiment will be "negative".

Location Extraction: To extract a user's location, we could rely on online mapping services (e.g. Google Maps, Yahoo! maps, etc.) or build custom geocoding services [41, 43]. More elaborate techniques could also be considered [18, 1], but these require a significant amount of data for each Twitter user, to build an accurate language profile.

In our system, we apply the technique proposed in [41]. More specifically, we create a local geocoding service, built on top of the GeoNames database [4]. We use Flickr data [5] to create an administrative hierarchy of places. The administrative hierarchy is necessary to discriminate between places with the same name, such as "Athens" in Georgia, US from "Athens" in Greece. Flickr provides data up to a town or suburb level, which we feel is more than sufficient for our purposes.

Along with every tweet, the Twitter service also sends information about the poster, such as the "Location" field of her profile. Tweets are mapped to locations as follows: First check whether it contains a GPS signal, which occurs when the tweet is automatically posted from a location based service (e.g. Foursquares). Otherwise, use the "Location" field and map the user with our custom gazetteer. Therefore, for that particular tweet, the user is assigned to the sensor that monitors the respective area. We do not consider tweets from users who can not be mapped to a location.

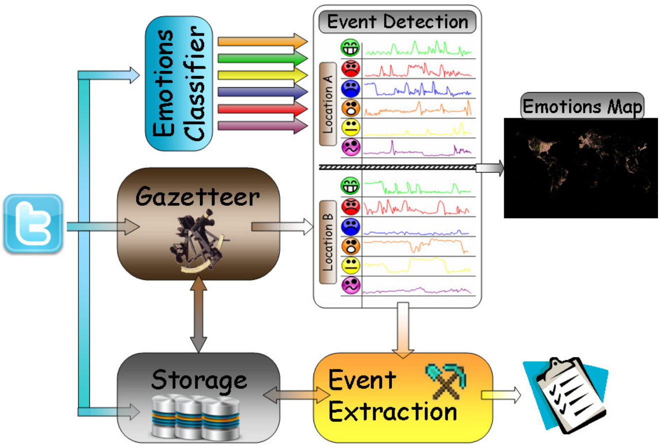


Figure 5: Schematic interaction of our system's components

Event Detection: We assume a set of virtual sensors, each of which is in charge of monitoring a specific location. Each sensor runs the event detection mechanism presented in Section 3, applied on the *counts* of each emotion, measured in every aggregation interval. Event detection runs in an *online* fashion, to address the "*as events occur*" requirement of our problem. When an event is detected in the emotional state of the users, a signal is raised, notifying the event extraction mechanism to be put into effect. The general idea is to run a lightweight, yet effective, mechanism for event detection, which will trigger the more heavyweight event extraction module when it should be executed, instead of constantly running the latter.

Event extraction: The event extraction mechanism is responsible for finding the most informative terms which describe the event that occurred. The input of this component is a set of tweets (e.g., their ids) that have been classified under the same emotion, and were observed during the aggregation interval that triggered the event. The output is a set of keywords which can effectively describe the event. Various techniques can be employed at this stage, ranging from simple ones, e.g., term frequency, TF-IDF, to more elaborate ones, such as similarity-based ones [10, 30, 7]. Given that the purpose of this module goes beyond the scope of this paper, we will not elaborate further.

It is important to point out that the same event may invoke different emotions. For example, a presidential election will be welcomed by some but will displease some others. This fact, which we are aware of, can be handled in a straightforward way, because such tweets will share a fair amount of common terms and, therefore, can be grouped together. Note that clustering based approaches will also fail, unless the tweets share some terms. The user can then be notified of the event and the emotions in an aggregate way (e.g., 60% happy, 30% angry, 10% sad). In fact, informing the end-user of the different emotions that an event may elicit adds a semantic dimension, which can be very helpful for social scientists and decision makers alike. This is a unique trait of our technique, not shared by any other existing approach.

4.1 Event Extraction Workflow

From the description of an event e and our event detection mechanism, it should be clear by now that we need the following infor-

[3] https://dev.twitter.com/

[4] http://geonames.org/

[5] http://www.flickr.com/places/info/

mation: 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. The standalone subsystems used to extract this information have already been described. Therefore, in the next paragraphs, we describe how we put the pieces of the puzzle together to serve our ultimate goal of event detection.

Figure 5 shows a schematic view of the components and their interaction[6]. The Twitter stream is our system's input, feeding two components, namely the *emotions classifier* and the *location extraction* subsystem. Through the location extraction process, each incoming tweet is mapped to a location, which will also be the location of the event (assuming one occurs). As discussed in Section 3, we use a set of virtual sensors, each of which is solely responsible for a specific location. Therefore, the tweet will be forwarded to the virtual sensor responsible for the location it was mapped to.

Meanwhile, the tweet has been classified to one of the 7 emotions that we use. If the classifier indicated that the tweet conveys a non-neutral emotion, the virtual sensor will further process it. Otherwise, no additional actions are performed. In any case, we store all tweets in our database. 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.

All in all, when a sensor receives a tweet for further processing, we already know its location and which of the 6 non-neutral emotions it has been cast to. For each emotion independently, the sensor counts how many tweets it has received during the last aggregation interval, and produces a single value (the respective count) at the end of each aggregation interval. Each such value serves as input to a separate instance of the event detection mechanism we have previously described, running on the same sensor. The event detection module updates its values and identifies whether a surge, i.e., an event, in any emotional state has occurred. In such a case, 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 particular emotion are passed on to the event extraction mechanism, which will go over them and summarize the event appropriately. This final step will provide the descriptive keywords of the event, which can then be presented to the user with all the necessary information: *location, timestamp, emotion* and *description*. Notice how the event extraction step is put into motion only if an event was detected. A sample output can be found in [42].

5. EXPERIMENTS

5.1 Experimental Setup

Dataset: We have a dataset of nearly 300 million tweets, between the start of April and end of May 2012, obtained through Twitter's Gardenhose. This gives us a 10% sample of all public tweets posted on Twitter. Therefore, we have no control over the tweets that we receive in terms of content, location, or language, nor the rates in which they arrive. Given that very few users changed their profile location during this period, we assume that their actual location is the one defined in the profile. We go over all tweets from users who have been mapped to a location and, using language detection software [17], ignore anything that is not: *english, spanish, german* or *greek*. We also only consider users from *Canada, France, Greece, Germany, Ireland, Spain, UK, US*. This gives us approximately 33.5M tweets, from a little less than 400K unique users. We order the tweets by their timestamp and replay the stream, feeding them to the event detection mechanism. We apply no other data cleaning, since we would like to perform our evaluation as it should occur in a real-life setup.

Techniques: We have implemented our system, *TwInsight*, and the state-of-the-art, EDCoW [44], in Java 1.6, both of which rely on Twitter alone as input. We run our experiments on a Quad Core machine, @3.5 GHz, with 16GB RAM. We report the average results of three runs.

Emotions Classification: To train our classifier, we asked from 30 individuals, of varied expertise on Twitter, to annotate a set of tweets with one of Ekman's emotions, or a "None" option, according to their belief that the tweet conveyed (or not) an emotion. They were also allowed to skip tweets they felt unsure about. The tweets were randomly selected from the initial dataset of 300M, to avoid biasing the evaluation, and no consesus phase took place.

This process created a gold standard of nearly 6700 tweets [7]. From these, we removed about 100 tweets that were outside the set of accepted languages, despite the Twitter users having indicated otherwise in their profiles. Taking our objective into account, we performed the following cleaning: For every tweet with an emotion and more than 100 retweets, we created an identical entry in the dataset with a random retweet count of up to 100. We then replaced the annotator's choice with "None" in the initial tweet and kept both versions. Most of these tweets were funny quotes and memes – a commonplace in Twitter –, and had the same id, thus they could be efficiently identified through simpler mechanisms (counting the id of a tweet), even if they refer to an ongoing event. Finally, we feel that a user *reporting* an event, will *tweet* about it in their own words, than search for an existing one and retweet it.

We experimented with various classifiers in Weka [17], including SVMs and decision trees. Selecting the most frequent class ("None") would give a $\sim$34% accuracy. In the end, a C4.5 decision tree returned the highest accuracy (64.39%), in 10-fold cross validation, with other classifiers yielding similar results. Clearly, these values can be improved, but one should also consider the subjective nature of the experiment. Most importantly, though, this did not prevent us from identifying meaningful events. Finally, techniques such as [19] are applicable, given the high discrepancy in sizes between our gold standar and the experimental dataset.

5.2 Effect of parameters

In this section we evaluate the effect of parameters of our kernel-based mechanism in the event identification process. In particular, we want to see how they affect the times a trigger is raised, putting the event extraction phase into motion. Although a high number of triggering might fulfill more users' needs, it also means that *TwInsight* is stressed more, especially if these are false positives.

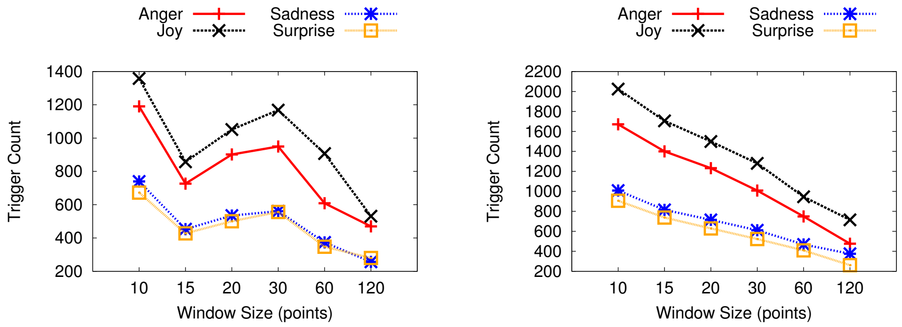
Table 1 shows the number of times that the event extraction phase was triggered, for various neighborhood ranges and sample sizes. We use 1 minute aggregation and maintain a window of 30 points, resulting in a history length of 30 minutes. We compare against an approach that relies on simple counting of received tuples ("Neutral"), instead of distinguishing between emotions.

Evidently, increasing the neighborhood range results in less triggers, because each point distributes its weight to a broader area. Therefore, new points receive more weight from previous ones, and are less likely to raise an event. Note that the actual numbers may seem quite high, but only account for less than 2.5% of all minutes in the 2 month period.

In addition to the semantic implications of distinguishing among tweets using emotions, this technique also appears to be a more lightweight approach when looking for events, as there is a 3×

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

[7] The dataset will be available upon request



(a) Sample size = 50%

(b) Sample size = 100%

Figure 6: #Times a trigger was raised, compared to the window size. a = 1min, r = 0.01, p=0.1

Table 1: #Times a trigger was raised, compared to the neighborhood range. a = 1min, w = 30, p = 0.1

	50% Sample		**100% Sample**	
Range	**Neutral**	**Joy**	**Neutral**	**Joy**
0.001	4977	1637	5315	1942
0.01	4266	1168	4138	1280
0.1	4198	1252	4088	1274

cut-down of triggering times. Also note that far fewer tuples are considered, because tweets classified as "None" are dropped. In other words, subsequent modules will be triggered fewer times and operate on fewer tweets, resulting in a reduced load overall.

5.3 SpatioTemporal Locality of Emotions

In this section we discuss the effect of spatio-temporal locality of emotions, and its importance in event detection. Figure 6 shows the number of times we raised an event as a function of the history we maintain, when aggregating emotions over the past 1 minute and monitoring the entire stream at once (we use only one sensor).

Interestingly, a bigger sample size results in more triggers. This is due to maintaining outdated information compared to the fast pace of the medium. As we increase the history length, triggering drops slowly (Figure 6(b)), because new points can be matched against more sampled data, and are less likely to be flagged as outliers. Recall that we monitor the entire stream here, implying that there is a continuous flow among all the regions we monitor.

On the other hand, Figure 6(a) leads to a very interesting observation. Using a 50% sample, there is a dramatic drop in the number of triggers, when we increase the window size from 10 to 15; from that point, until a window size of 30, triggering events increases slightly, and begins decreasing from that point on. This means that for 1 minute aggregations, there are rapid changes in the observed emotions; therefore a window of 10 points may be too narrow, to maintain a representative "history". On the other hand, a window between 15 – 30 minutes seems like a better choice. This result correlates very well with the real time nature of the medium, where people tend to speak and respond very quickly to their tweets. It also means that events that are present in our data create some momentum over a mid-size period ($\sim$30minutes), and then dissipate.

Given that 1-minute aggregations may be too aggressive, we also experimented with 5-minute aggregations. As shown in Figure 7 the 5-minutes aggregations are smoother, when using the 50% sample. Moreover, triggering is increased towards the end, where the window size is at least 24 points, i.e. 2hours. However, most triggers did not correspond to particularly meaningful events using simple term frequency for event description. This implies that there are not such sudden changes in 5-minute aggregations, when monitoring the entire stream. This has also been validated with longer aggregation periods.

What is more interesting, however, is the spike we observe in Figure 7(b), for a window size of 15 points (75 minutes), which does not exhibit a similar behavior with the one of Figure 6(a). The explanation is given in Figure 7(c), where we plot the emotion of *joy* for Canada compared to the average of the rest of the countries. The discrepancy between the two lines for w=12 and w=15 is the cause of the spike shown in Figure 7(b). In practice, this means that emotions are also exhibit spatial coherency. Therefore, it is better to perform localized monitoring, rather than bluntly looking into the entire stream at once.

5.4 Efficiency Performance

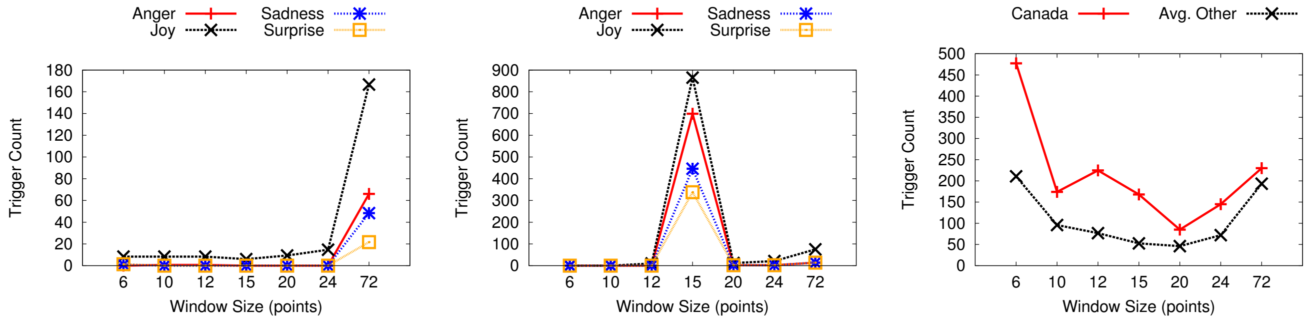
In the following paragraphs, we compare *TwInsight* in terms of efficiency and effectiveness against *EDCoW* [44], the current state-of-the-art for event detection in Twitter.

Table 2 summarizes for both systems the average time taken by each component to apply its functionality on a newly received tuple. Location extraction is a common component. Therefore, any differences lie in the classification and event detection steps.

Table 2: Average Component Processing Time (ms)

Module	**TwInsight**	**EDCoW**
Location Extraction	3.36	
Classification	0.35	-
Event Detection	0.001	20 – 200
Total	3.72	23.36 – 203.36

We achieve a total 6× speed-up compared to *EDCoW*, with the event detection step being orders of magnitude faster. In other words, our approach requires considerably less time to identify events, and is expected to scale better as the number of tweets increases. The reason is that *TwInsight* operates in an online fashion, and new tuples are processed only once upon arrival. On the contrary, *EDCoW* requires between 20 and 200ms, depending on the value of Δ, which specifies the number of Stage-2 signals. This difference in efficiency is the result of several reasons: i) *EDCoW* revisits tuples to produce the Stage-2 signal values, ii) computes the wavelet coefficients in all levels, which is a function of Δ, iii) computes signals' autocorrelations – an unavoidable cost, iv) ap-



(a) Sample Size = 50%

(b) Sample Size = 100%

(c) joy in Canada

Figure 7: #Times a trigger was raised, compared to the window size. a = 5min, r = 0.01, p=0.1

plies the *median absolute deviation* filter twice and *iii*) performs the clustering step each time anew.

We would also like to note that we have run *TwInsight* on a simple laptop with 2GB RAM. On the contrary, *EDCoW* can not operate with so little memory, as it must maintain (at least) the stage-1 signals for every word it has encountered.

5.5 Event Detection Effectiveness

To evaluate the effectiveness of the techniques, we compare their ability to identify significant events from our dataset. Obviously, neither technique can identify events which are not mentioned at all. For this reason, we present significant events that each technique extracted, cross validating them with online resources.

EDCoW: We initially run *EDCoW* with the parameters suggested by the authors, i.e., 10 minute intervals for stage-1 signals, $\Delta = 6$, $\gamma = 40$, over the duration of one day. As described in [44], an event *must contain at least two terms, but not too many*. Note that, for these parameters, the average description length of an event in their dataset was 2.23 terms, with at most 3 tokens.

In our dataset, *EDCoW* finds 709 events with at least 2 keywords. However, the average event description is 39.7 terms (σ = 26.6). Therefore, we set $\gamma = 200$, to increase the number of filtered tokens. Although this is 5× higher than what the authors originally used, the average description remains high: 21 terms ($\sigma = 23.2$). In other words, most of the identified events would have been filtered out due to low significance.

Regardless, we searched for terms that we knew to have occurred during our crawl. The terms "eurovision" (the contest), "bayern" and "drogba" (for the Champions League final), did not return any results. The term "chelsea", – the other competing team –, appears in 4 events, when using $\gamma = 40$; the term does not appear when $\gamma = 200$. Most importantly, though, the term appears on May 24 2012, 5 days *after* the match. One of those 4 cases is shown below. Clearly, most of the terms are unrelated:

bobcat, liar, desir, doc, chainz, push, selena, quedo, pasado, understood, gustan, howard, older, rare, technolog, **chelsea***, stadium, phoenix, fit, concern, psalm, thug, duda, pacer, irish, hah, hacerlo, provid, debat, swag, mum, pregunto, lama, vou, lux, strike, swallow, cuerpo, grow, goal, theori, singer, yung, lookin, 500, slapen, lea, suspens, 2016, ignor, marri*

Given our goal to identify events in a timely fashion, we also experimented with 1 minute aggregations, $\Delta = 5$, $\Gamma = 40$, over the duration of half an hour, similarly to our parameters. This gives 588 events, with an average description length of 6 tokens, but is filled with cusses. The only exception of a real-life event is the case of "don,celtic" identified on May 27, 00:27 GMT. The event most likely refers to the Boston Celtics game, which took place on May 26, 8:00 ET (i.e., May 27, 00:00 GMT).

All in all, *EDCoW* failed to identify meaningful events that we knew about during this period. We identify two key reasons: *a*) Unlike [44], our dataset was processed as received, without applying any cleaning that would ensure adequate volume of the remaining tokens, *b*) the clustering step is prone to the data cleaning process and groups tokens together aggressively. This can be validated by the need for EDCoW's *event significance* step, to prune clusters with too many terms.

TwInsight: On the contrary, *TwInsight* identified several events during this period, and Table 3 presents a brief summary of the most prominent, along with the associated emotion, where and when it was discovered, given our objective to identify events *as they occur*.

Event 1 concerns the speech that Erskine Bowles, a renowned figure in the US, gave to the graduates of an american university on that day. Given that posted excerpts contained terms such as "debt", and "crazy", they were flagged with the emotion of "Fear".

Event 2 is about an australian company that failed to raise a required amount of $150m, to finish the construction of the Ararat prison. The company went under voluntary administration [8].

Event 3 can be traced back to various NBA games occurring on *May 15*, but are observed on *May 16* due to the time difference. Both "Spurs" and "Heat" played on May 15 ("Spurs" when the event was detected), and "OKC" had a game the following day. Note that these games appear together because we report events at the country level, rather than the city level.

Event 4 refers to the "Blockupy Frnakfurt" movement in Germany on May 16/05.

Event 5 is about a major event in our dataset, the Champions League (CL) 2012 finals, between Bayern and Chelsea, that took place on May 19. The game began at 20:45 CEST (18:45 GMT), and there are supportive tweets for the teams just as the match was about to begin.

Event 6 is related to Chris Kreider, a NY Rangers (NYR) Hockey player, who scored a goal in 5' 16" of the 3rd period. A regular hockey game has three 20' periods, with 2 intermissions of 17' each. The game began at 18:00 GMT (13:00 ET [9]), placing the goal no sooner than 19:19 GMT, not accounting for any delays, and certainly no later than 19:27, when we identify it.

Event 7 is related to the goal by Bayern's football player, Thomas Müller, in the CL final. The goal was scored in the 83rd minute of

[8] http://www.theaustralian.com.au/business/property/st-hilliers-arafat-arm-fails-to-secure-150m-goes-into-administration/story-fn9656lz-1226357382452

[9] http://www.nhl.com/ice/recap.htm?id=2011030313

Table 3: Sample Summary of 15 Prominent Events Identified By *TwInsight*

ID	Emotion	Where	When (GMT)	Description
1	Fear	US	13/05, 13:53	bowles crazy crisis debt national single spent year ẽrskine #2012AUGrad
2	Surprise	Canada	15/05, 23:42	$150m administration ararat couldn due extra funding hilliers
3	Sadness	Canada	16/05, 2:20	spade heat okc ship sigh spurs win @luggageboii @monalove810 calling
4	Anger	Germany	16/05, 07:52	aus frankfurt live radio @eThn0 http://t.co/ij9MNILL occupy
5	Joy	UK	19/05, 18:37	bayern win based champions chelsea excited fair fan final germans
6	Joy	UK	19/05, 19:27	15th 3rd 5th chris game games goal kreider nyr wel
7	Joy	Germany	19/05, 20:23	thomas bayern championsleague cfc mueller muller müller
8	Joy	UK	19/05, 20:29	didier drogba fucking beauty enjoying fair gal gaz goal great
9	Sadness	Canada	20/05, 22:42	died bich breaking mio robin singer @rodneyedwards gib gibb opa
10	Anger	Canada	20/05, 23:04	nose didnt live nato plz police protestors riot tryna tweet
11	Anger	Canada	20/5, 15:19	@ctvcalgary aime ambition chacun earthquake femme frais http://t.co/0hJEez9Q italy kills
12	Anger	US	20/5, 11:23	@Mou2amara alive assad onus prove regime shawkat showusshaukat syria
13	Anger	Ireland	26/05, 19:11	fahey ireland jesus keith paul squad suck tlist caled
14	Anger	Greece	26/05, 19:52	eurovision rain too much #Eurovision2012 kuulaaaaaaa
15	Joy	US	27/05, 00:58	celtics comin days game left #IRELAND #SOEXCITED @NICKIMINAJ looking

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

Event 9 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 [10], and a surge in sad tweets is seen at 22:42, only 10' after his death.

Event 10 concerns the riots in Chicago, where protestors were opposing Nato's Summit [11]. Due to words "protestors", "opposing", "riot", the conveyed feeling is anger.

Event 11 is about the earthquake in Italy, on May 20, that resulted in the death of six people, among them a woman.

Event 12 refers to Assef Shawkat, deputy Minister of Defense of Syria. On May 20, 2012, there was a claim he had been murdered [12], 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.

Event 13 is about Keith Fahey, an Irish football player, who was injured and pulled out of the national team.

Event 14 is one of many regarding the Eurovision contest, which took place on May 26. Most of them are related with Joy, however, the one we show here is related with anger. The original tweets are in greek and we show the transliteration. The day of the contest was a rainy and the phrase "too much" indicates the posters' dislike of both the contest and the fact that it was raining. Moreover, *Kuula* was the name of the Estonian song, but also happens to be a Greek name, popularized by an 80's greek comedy, to convey anguish. Moreover, the tweets we have identified follow the sequence in which the songs were performed. For instance, the tweets of this event appear after tweets containing "eurovision" and "Italy", which is expected, as Estonia performed right after Italy. Finally, the contest started at 21:00 CET [13] (19:00 GMT), and the Estonian song was performed 52 minutes within the contest [14]. Therefore, we identify the event as timely as possible.

Event 15 is about the NBA game between Boston Celtics and the 76ers. The game started at 20:00 ET, *26 May* 2012, which is 1:00 am GMT, **May 27th**. Discussions on Twitter prior to the game, especially as it was about to start, led to the event being detected. There were also some feelings of excitement concerning Nicki Minaj's upcoming performance in Ireland.

To sum up, *TwInsight* identified several events of varying types, emotions, and intensity, even though we use simpler techniques for event description. Note that we also identified different events happening on the same day, because we can operate effectively in smaller time intervals. Moreover, depending on its type, an event's transition from a latent state to one where it has gained enough visibility may vary, nevertheless, we are still able to capture such changes. Therefore, we have been able to identify considerably more meaningful events, with more efficient techniques, compared with *EDCoW*. Finally, it is worth noting that, even though we presented the locations where the events were observed, we are able to identify most of them even when we monitor the entire stream.

6. 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 its superiority over the state-of-the-art.

We plan to improve our work on several fronts, e.g., i) performance of the location extraction method, by applying online location clustering, using GPS signals, ii) event description / summarization technique, iii) classification accuracy to filter uninformative points even further, etc. We also plan conduct an analysis over a more extended period of time.

Acknowledgements: The authors would like to thank the data annotators. This work has been co-financed by EU and Greek

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

[11] http://www.huffingtonpost.com/2012/05/20/nato-summit-chicago-protesters_n_1530789.html

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

[13] http://www.eurovision.tv/page/baku-2012/about/shows

[14] http://www.youtube.com/watch?v=fjue0I4Hyko#t=51m40s

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.

7. REFERENCES

[1] A. Ahmed, L. Hong, and A. J. Smola. Hierarchical geographical modeling of user locations from social media posts. WWW, 2013.

[2] F. Alvanaki, S. Michel, K. Ramamritham, and G. Weikum. See what's enblogue: real-time emergent topic identification in social media. In *EDBT*, 2012.

[3] B. Babcock, M. Datar, and R. Motwani. Sampling from a moving window over streaming data. In *SODA*, 2002.

[4] L. Barbosa and J. Feng. Robust sentiment detection on twitter from biased and noisy data. COLING, 2010.

[5] H. Becker, F. Chen, D. Iter, M. Naaman, and L. Gravano. Automatic identification and presentation of twitter content for planned events. In *ICWSM*, 2011.

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

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

[8] E. Benson, A. Haghighi, and R. Barzilay. Event discovery in social media feeds. In *ACL-HLT*, 2011.

[9] J. Bollen, H. Mao, and A. Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *ICWSM*, 2011.

[10] D. Chakrabarti and K. Punera. Event summarization using tweets. In *ICWSM*, 2011.

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

[12] P. Ekman, W. Friesen, and P. Ellsworth. *Emotion in the human face: guide-lines for research and an integration of findings*. Pergamon Press, 1972.

[13] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *Processing*, 2009.

[14] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: the underground on 140 characters or less. In *CCS*, 2010.

[15] D. Gunopulos, G. Kollios, V. J. Tsotras, and C. Domeniconi. Approximating multi-dimensional aggregate range queries over real attributes. SIGMOD, 2000.

[16] M. Gupta, P. Zhao, and J. Han. Evaluating event credibility on twitter. In *SDM*, 2012.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1), nov 2009.

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

[19] N. Katariya, A. Iyer, and S. Sarawagi. Active evaluation of classifiers on large datasets. In *ICDM*, 2012.

[20] J. Kleinberg. Bursty and hierarchical structure in streams. In *SIGKDD*, 2002.

[21] T. Lansdall-Welfare, V. Lampos, and N. Cristianini. Effects of the recession on public mood in the uk. In *WWW Companion*, 2012.

[22] T. Lappas, M. R. Vieira, D. Gunopulos, and V. J. Tsotras. On the spatiotemporal burstiness of terms. *PVLDB*, 5(9), 2012.

[23] G. Leshed and J. J. Kaye. Understanding how bloggers feel: recognizing affect in blog posts. In *CHI*, 2006.

[24] M. Lewis, J. Haviland-Jones, and L. Barrett. *Handbook of Emotions, Third Edition*. Guilford Publications, 2010.

[25] C. Li, A. Sun, and A. Datta. Twevent: segment-based event detection from tweets. In *CIKM*, 2012.

[26] B. M. Luisa Bentivogli, Pamela Forner and E. Pianta. Revising wordnet domains hierarchy: Semantics, coverage, and balancing. COLING, 2004.

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

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

[29] G. Mishne and M. de Rijke. Capturing global mood levels using blog posts. In *AAAI-CAAW*, 2006.

[30] J. Nichols, J. Mahmud, and C. Drews. Summarizing sporting events using twitter. In *IUI*, 2012.

[31] M. Osborne, S. Petrovic, R. McCreadie, C. Macdonald, and I. Ounis. Bieber no more: First story detection using twitter and wikipedia.

[32] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2), jan 2008.

[33] J. W. Pennebaker and L. A. King. Linguistic styles: language use as an individual difference. *Journal of personality and social psychology*, 77(6), 1999.

[34] A. Ritter, Mausam, O. Etzioni, and S. Clark. Open domain event extraction from twitter. SIGKDD, 2012.

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

[36] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. Twitterstand: news in tweets. In *SIGSPATIAL-GIS*, 2009.

[37] D. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1992.

[38] D. A. Shamma, L. Kennedy, and E. F. Churchill. Tweet the debates: understanding community annotation of uncollected sources. In *WSM*, 2009.

[39] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos. Online outlier detection in sensor data using non-parametric models. VLDB, 2006.

[40] 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.

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

[42] G. Valkanas and D. Gunopulos. A ui prototype for emotion-based event detection in the live web. In *CHI-KDD*, pages 89–100, 2013.

[43] J. Vosecky, D. Jiang, and W. Ng. Limosa: a system for geographic user interest analysis in twitter. In *EDBT*, 2013.

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