HAL
open science

# Event detection, tracking and visualization in Twitter A mention-anomaly-based approach

Adrien Guille, Cécile Favre

## To cite this version:

Adrien Guille, Cécile Favre. Event detection, tracking and visualization in Twitter A mention-anomaly-based approach. Social Network Analysis and Mining, 2015, 5 (1), pp.art. 18. 10.1007/s13278-015-0258-0 . hal-01158068

**HAL Id: hal-01158068**

**https://hal.science/hal-01158068v1**

Submitted on 29 May 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**Social Network Analysis and Mining**
DOI 10.1007/s13278-015-0258-0 The final publication is available at www.springerlink.com

# Event detection, tracking and visualization in Twitter

## A mention-anomaly-based approach

**Adrien Guille · Cécile Favre**

Received: December 14, 2014 / Revised: April 3, 2015 / Accepted: May 13, 2015

**Abstract** The ever-growing number of people using Twitter makes it a valuable source of timely information. However, detecting events in Twitter is a difficult task, because tweets that report interesting events are overwhelmed by a large volume of tweets on unrelated topics. Existing methods focus on the textual content of tweets and ignore the social aspect of Twitter. In this paper we propose *MABED* (mention-anomaly-based Event Detection), a novel statistical method that relies solely on tweets and leverages the creation frequency of dynamic links (*i.e.* mentions) that users insert in tweets to detect significant events and estimate the magnitude of their impact over the crowd. *MABED* also differs from the literature in that it dynamically estimates the period of time during which each event is discussed, rather than assuming a predefined fixed duration for all events. The experiments we conducted on both English and French Twitter data show that the mention-anomaly-based approach leads to more accurate event detection and improved robustness in presence of noisy Twitter content. Qualitatively speaking, we find that *MABED* helps with the interpretation of detected events by providing clear textual descriptions and precise temporal descriptions. We also show how *MABED* can help understanding users' interest. Furthermore, we describe three visualizations designed to favor an efficient exploration of the detected events.

**Keywords** Event detection · Twitter · Visualization

A. Guille
ERIC Lab, University of Lyon 2
5 avenue Pierre Mends France
69676 Bron, France
E-mail: adrien.guille@univ-lyon2.fr

C. Favre
cecile.favre@univ-lyon2.fr

## 1 Introduction

Twitter is a social networking and micro-blogging service that allows users to publish short messages limited to 140 characters, *i.e.* tweets. Users share, discuss and forward various kinds of information – ranging from personal daily events to important and global event related information – in real-time. The ever-growing number of users around the world tweeting makes Twitter a valuable source of timely information. On the other hand, it gives rise to an information overload phenomenon and it becomes increasingly difficult to identify relevant information related to *significant events*. An event is commonly defined as a thing that happens at one specific time (Becker et al, 2011; Aggarwal and Subbian, 2012), and it is significant if it may be discussed by traditional media (McMinn et al, 2013). These facts raise the following question: *How can we use Twitter for automated significant event detection and tracking?* The answer to this question would help analyze which events, or types of events, most interest the crowd. This is critical to applications for journalistic analysis, playback of events, *etc.* Yet the list of "trends" determined by Twitter isn't so helpful since it only lists isolated keywords and provides no information about the level of attention it receives from the crowd nor temporal indications.

Twitter delivers a continuous stream of tweets, thus allowing the study of how topics grow and fade over time (Yang and Leskovec, 2011). In particular, event detection methods focus on detecting "bursty" patterns – which are intuitively assumed to signal events (Kleinberg, 2002) – using various approaches ranging from term-weighting-based approaches (Shamma et al, 2011; Benhardus and Kalita, 2013) to topic-modeling-based approaches (Lau et al, 2012; Yuheng et al, 2012), includ-

ing clustering-based approaches (Weng and Lee, 2011; Li et al, 2012; Parikh and Karlapalem, 2013). Despite the wealth of research in the area, the vast majority of prior work focuses on the textual content of tweets and mostly neglects the social aspect of Twitter. However, users often insert extra-textual content in their tweets. Of particular interest is the "mentioning practice", which consists of citing other users' screennames in tweets (using the syntax "@username"). Mentions are in fact dynamic links created either intentionally to engage the discussion with specific users or automatically when replying to someone or re-tweeting. This type of link is dynamic because it is related to a particular time period, *i.e.* the tweet lifespan, and a particular topic, *i.e.* the one being discussed.

**Proposal** We tackle the issue of event detection and tracking in Twitter by devising a new statistical method, named *MABED* (mention-anomaly-based Event Detection). It relies solely on statistical measures computed from tweets and produces a list of events, each event being described by (i) a main word and a set of weighted related words, (ii) a period of time and (iii) the magnitude of its impact over the crowd. In contrast with existing methods, *MABED* doesn't only focus on the textual content of tweets but also leverages the frequency at which users interact through mentions, with the aim to detect more accurately the most impactful events. It also differs from the literature in that it dynamically estimates the period of time during which each event is discussed, rather than assuming a predefined fixed duration for all events, in order to provide clearer event descriptions. What is more, we develop three interactive visualizations to ensure an efficient exploration of the detected events: (i) a timeline that allows exploring events through time, (ii) a chart that plots the magnitude of impact of events through time and (iii) a graph that allows identifying semantically related events. The implementation of *MABED* is available for re-use and future research. It is also included in *SONDY* (Guille et al, 2013), an open-source social media data mining software that implements several state-of-the-art algorithms for event detection.

**Results** We perform quantitative and qualitative studies of the proposed method on both English and French Twitter corpora containing respectively about 1.5 and 2 millions tweets. We show that *MABED* is able to extract an accurate and meaningful retrospective view of the events discussed in each corpus, with short computation times. To study precision and recall, we ask human annotators to judge whether the detected events are meaningful and significant events. We demonstrate the relevance of the mention-anomaly-based approach, by showing that *MABED* outperforms a variant that ignores the presence of mentions in tweets. We also show that *MABED* advances the state-of-the-art by comparing its performance against those of two recent methods from the literature. The analysis of these results suggests that considering the frequency at which users interact through mentions leads to more accurate event detection and improved robustness in presence of noisy Twitter content. Lastly, we analyze the types of events detected by *MABED* with regard to the communities detected in the network structure (*i.e.* the following relationships) that interconnects the authors of the tweets. The results of this analysis shed light on the interplay between the social and topical structures in Twitter and show that *MABED* can help understanding users' interests.

The rest of this paper is organized as follows. In the next section we discuss related work, before describing in detail the proposed method in Section 3. Then an experimental study showing the method's effectiveness and efficiency is presented in Section 4. Next, we present three visualizations for exploring the detected events. Finally, we conclude and discuss future work in Section 6.

## 2 Related Work

Methods for detecting events in Twitter rely on a rich body of work dealing with event, topic and burst detection from textual streams. In a seminal work, Kleinberg (2002) studies time gaps between messages in order to detect bursts of email messages. Assuming that all messages are about the same topic, he proposes to model bursts with hidden Markov chains. AlSumait et al (2008) propose *OLDA* (On-line Latent Dirichlet Allocation), a dynamic topic model based on *LDA* (Blei et al, 2003). It builds evolutionary matrices translating the evolution of topics detected in a textual stream through time, from which events can be identified. Fung et al (2005) propose to detect and then cluster bursty words by looking at where the frequency of each word in a given time window is positioned in the overall distribution of the number of documents containing that word.

Tweet streams differ from traditional textual document streams, in terms of publishing rate, content, *etc.* Therefore, developing event detection methods adapted to Twitter has been studied in several papers in recent years. Next, we give a brief survey of the proposed approaches.

## 2.1 Event Detection and Tracking from Tweets

**Term-weighting-based approaches** The *Peakiness Score* (Shamma et al, 2011) is a normalized word frequency metric, similar to the $tf \cdot idf$ metric, for identifying words that are particular to a fixed length time window and not salient in others. However, individual words may not always be sufficient to describe complex events because of the possible ambiguity and the lack of context. To cope with this, Benhardus and Kalita (2013) propose a different normalized frequency metric, *Trending Score*, for identifying event-related n-grams. For a given n-gram and time window, it consists in computing the normalized frequency, $tf_{norm}$, of that n-gram with regard to the frequency of the other n-grams in this window. The *Trending Score* of a n-gram in a particular time window is then obtained by normalizing the value of $tf_{norm}$ in this time window with regard to the values calculated in the others.

**Topic-modeling-based approaches** Lau et al (2012) propose an online variation of *LDA*. The idea is to incrementally update the topic model in each time window using the previously generated model to guide the learning of the new model. At every model update, the word distribution in topics evolves. Assuming that an event causes a sudden change in the word distribution of a topic, authors propose to detect events by monitoring the degree of evolution of topics using the Jensen-Shannon divergence measure. Yuheng et al (2012) note that topic modeling methods behave badly when applied to short documents such as tweets. To remedy this, they propose *ET-LDA* (joint Event and Tweets *LDA*). It expands tweets with the help of a search engine and then aligns them with re-transcriptions of events provided by traditional media, which heavily influences the results. Globally, topic-modeling-based methods suffer from a lack of scalability, which renders their application to tweet streams difficult. However, works by Aiello et al (2013) reveal that dynamic topic models don't effectively handle social streams in which many events are reported in parallel.

**Clustering-based approaches** *EDCoW* (Weng and Lee, 2011) breaks down the frequency of single words into wavelets and leverages Fourier and Shannon theories to compute the change of wavelet entropy to identify bursts. Trivial words are filtered away based on their corresponding signal's auto correlation, and the similarity between each pair of non-trivial words is measured using cross correlation. Eventually, events are defined as bags of words with high cross correlation during a predefined fixed time window, detected with modularity-based graph clustering. However, as pointed out by Li et al (2012) and Parikh and Karlapalem (2013), measuring cross correlation is computationally expensive. Furthermore, measuring similarity utilizing only cross correlation can result in clustering together several unrelated events that happened in the same time span. *TwEvent* (Li et al, 2012) detects event from tweets by analyzing n-grams. It filters away trivial n-grams based on statistical information derived from Wikipedia and the Microsoft Web N-Gram service. The similarity between each pair of non-trivial n-grams is then measured based on frequency and content similarity, in order to avoid merging distinct events that happen concurrently. Then, similar n-grams in fixed-length time windows are clustered together using a $k$-nearest neighbor strategy. Eventually, the detected events are filtered using, again, statistical information derived from Wikipedia. As a result, the events detected with *TwEvent* are heavily influenced by Microsoft Web N-Gram and Wikipedia, which could potentially distort the perception of events by Twitter users and also give less importance to recent events that are not yet reported on Wikipedia. It is also worth mentioning *ET* (Parikh and Karlapalem, 2013), a recent method similar to *TwEvent*, except that it doesn't make use of external sources of information and focuses on bigrams. The similarity between pairs of bigrams is measured based on normalized frequency and content similarity, and the clustering is performed using a hierarchical agglomerative strategy.

## 2.2 Event Visualization in Twitter

*Eddi* (Bernstein et al, 2010) is among the first tools developed for visualizing events from tweets. It displays a single word cloud that describes all the detected events, as well as a single stacked area chart that plots the evolution of the relative volume of tweets for each event. Mathioudakis and Koudas (2010) propose *TwitterMonitor*, a system that allows for a finer understanding of the detected events in comparison with *Eddi*. It displays a list of events, each event being described by a set of words and a chart that plots the evolution of the volume of related tweets. *KeySEE* (Lee et al, 2013) is a tool that offers similar functionalities with more sophisticated visualizations, such as word clouds to describe events instead of sets of words. Marcus et al (2011) describe *TwitInfo*, a tool whose interface revolves around a timeline of events. The user can click an event to see the tweets published during the related time interval, or to see the most cited URLs in these tweets. Let us also mention work by Kraft et al (2013), in which a heatmap describes the distribution of events across time.

**Table 1** Table of notations.

| Notation | Definition |
|---|---|
| $N$ | Total number of tweets in the corpus |
| $N^i$ | Number of tweets in the $i^{\text{th}}$ time-slice |
| $N^i_t$ | Number of tweets in the $i^{\text{th}}$ time-slice that contain the word $t$ |
| $N_{@t}$ | Number of tweets in the corpus that contain the word $t$ and at least one mention |
| $N^i_{@t}$ | Number of tweets that contain the word $t$ and at least one mention in the $i^{\text{th}}$ time-slice |

## 3 Proposed Method

In this section, we first formulate the problem we intend to solve. Then we give an overview of the solution we propose, *MABED*, before describing it formally.

### 3.1 Problem Formulation

**Input** We are dealing with a tweet corpus $\mathcal{C}$. We discretize the time-axis by partitioning the tweets into $n$ time-slices of equal length. Let $V$ be the vocabulary of the words used in all the tweets and $V_@$ be the vocabulary of the words used in the tweets that contain at least one mention. Table 1 gives the definitions of the notations used in the rest of this paper.
**Output** The objective is to produce a list $L$, such that $|L| = k$, containing the events with the $k$ highest magnitude of impact over the crowd's tweeting behavior. We define an event as a bursty topic, with the magnitude of its impact characterized by a score. Definitions 1 and 2 below respectively define the concepts of bursty topic and event.

**Definition 1 (Bursty Topic)** Given a time interval $I$, a topic $T$ is considered bursty if it has attracted an uncommonly high level of attention (in terms of creation frequency of mentions) during this interval in comparison to the rest of the period of observation. The topic $T$ is defined by a main term $t$ and a set $S$ of weighted words describing it. Weights vary between 0 and 1. A weight close to 1 means that the word is central to the topic during the bursty interval whereas a weight closer to 0 means it is less specific.

**Definition 2 (Event)** An event $e$ is characterized by a bursty topic $\mathcal{BT} = [T, I]$ and a value $Mag > 0$ indicating the magnitude of the impact of the event over the crowd.

### 3.2 Overview of the Proposed Method

The method has a two-phase flow. It relies on three components: (i) the detection of events based on mention-anomaly, (ii) the selection of words that best describe each event and (iii) the generation of the list of the $k$ most impactful events. The overall flow, illustrated on Figure 1, is briefly described hereafter.

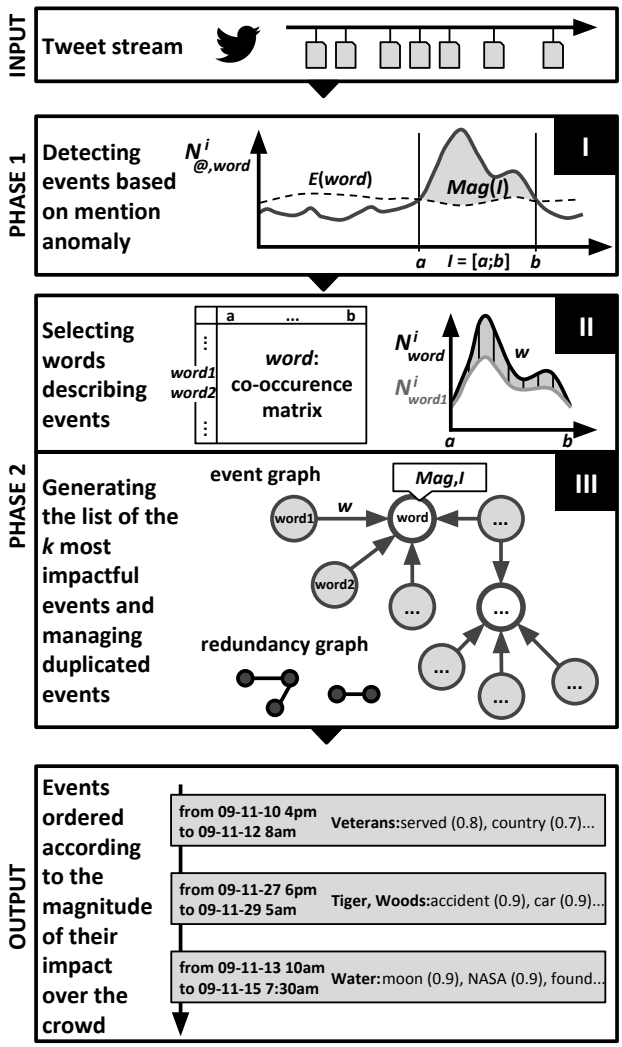



**Fig. 1** Overall flow of the proposed method, *MABED*.

1. The mention creation frequency related to each word $t \in V_@$ is analyzed with the first component. The result is a list of partially defined events, in that they are missing the set $S$ of related words. This list is ordered according to the impact of the events.
2. The list is iterated through starting from the most impactful event. For each event, the second component selects the set $S$ of words that best describe it. The selection relies on measures based on the co-occurrence and the temporal dynamics of words tweeted during $I$. Each event processed by this component is then passed to the third component, which

is responsible for storing event descriptions and managing duplicated events. Eventually, when $k$ distinct events have been processed, the third component merges duplicated events and returns the list $L$ containing the top $k$ events.

### 3.3 Detection of Events Based on Mention Anomaly

The objective of this component is to precisely identify when events happened and to estimate the magnitude of their impact over the crowd. It relies on the identification of bursts based on the computation of the anomaly in the frequency of mention creation for each individual word in $V_@$. Existing methods usually assume a fixed duration for all events that corresponds to the length of a time-slice. It's not the case with *MABED*. In the following, we describe how to compute the anomaly of a word for a given time-slice, then we describe how to measure the magnitude of impact of a word given a contiguous sequence of time-slices. Eventually, we show how to identify the intervals that maximize the magnitude of impact for each word in $V_@$.

**Computation of the anomaly at a point** Before formulating the anomaly measure, we define the expected number of mention creation associated to a word $t$ for each time-slice $i \in [1; n]$. We assume that the number of tweets that contain the word $t$ and at least one mention in the $i^{\text{th}}$ time-slice, $N^i_{@t}$, follows a generative probabilistic model. Thus we can compute the probability $P(N^i_{@t})$ of observing $N^i_{@t}$. For a large enough corpus, it seems reasonable to model this kind of probability with a binomial distribution (Fung et al, 2005). Therefore we can write:

$$P(N^i_{@t}) = \binom{N^i}{N^i_{@t}} p_{@t}^{N^i_{@t}} (1 - p_{@t})^{N^i - N^i_{@t}}$$

where $p_{@t}$ is the expected probability of a tweet containing $t$ and at least one mention in any time-slice. Because $N^i$ is large we further assume that $P(N^i_{@t})$ can be approximated by a normal distribution (Li et al, 2012), that is to say:

$$P(N^i_{@t}) \sim \mathcal{N}(N^i p_{@t}, N^i p_{@t}(1 - p_{@t}))$$

It follows that the expected frequency of tweets containing the word $t$ and at least one mention in the $i^{\text{th}}$ time-slice is:

$$E[t|i] = N^i p_{@t} \text{ where } p_{@t} = N_{@t}/N$$

Eventually, we define the anomaly of the mention creation frequency related to the word $t$ at the $i^{\text{th}}$ time-slice this way:

$$anomaly(t, i) = N^i_{@t} - E[t|i]$$

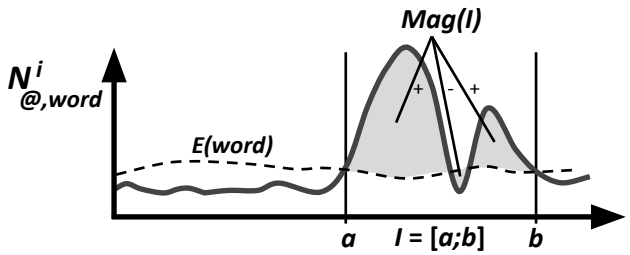



**Fig. 2** Identification of the time interval that maximizes the magnitude of impact for a given word.

With this formulation, the anomaly is positive only if the observed mention creation frequency is strictly greater than the expectation. Event-related words that are specific to a given period of time are likely to have high anomaly values during this interval. In contrast, recurrent (*i.e.* trivial) words that aren't event-specific are likely to show little discrepancy from expectation. What is more, as opposed to more sophisticated approaches like modeling frequencies with Gaussian mixture models, this formulation can easily scale to the number of distinct words used in tweets.

**Computation of the magnitude of impact** The magnitude of impact, *Mag*, of an event associated with the time interval $I = [a; b]$ and the main word $t$ is given by the formula below. It corresponds to the algebraic area of the anomaly function on $[a; b]$.

$$Mag(t, I) = \int_a^b anomaly(t, i)\,\mathrm{d}i = \sum_{i=a}^{b} anomaly(t, i)$$

The algebraic area is obtained by integrating the discrete anomaly function, which in this case boils down to a sum.

**Identification of events** For each word $t \in V_@$, we identify the interval $I$ that maximizes the magnitude of impact, that is to say :

$$I = \underset{I}{\operatorname{argmax}}\ Mag(t, I)$$

Because the magnitude of impact of an event described by the main word $t$ and the time interval $I$ is the sum of the anomaly measured for this word over $I$, this optimization problem is similar to a "Maximum Contiguous Subsequence Sum" (*MCSS*) problem. The *MCSS* problem is well known and finds application in many fields (Fan et al, 2003; Lappas et al, 2009). In other words, for a given word $t$ we want to identify the interval $I = [a; b]$, such that:

$$Mag(t, I) = \max\{\sum_{i=a}^{b} anomaly(t, i) | 1 \leqslant a \leqslant b \leqslant n\}$$

This formulation permits the anomaly to be negative at some points in the interval, as shown in Figure 2, only if it permits extending the interval while increasing the total magnitude. This is a desirable property, as it avoids fragmenting events that last several days because of the lower activity on Twitter during the night for instance, which can lead to low or negative anomaly. Another desirable property of this formulation is that a given word can't be considered as the main word of more than one event. This increases the readability of events for the following reason. The bigger the number of events that can be described by a given word, the less specific to each event this word is. Therefore, this word should rather be considered as a related word than the main word. We solve this *MCSS* type of problem using the linear-time algorithm described by Bentley (1984). Eventually, each event detected following this process is described by: (i) a main word $t$ (ii) a period of time $I$ and (iii) the magnitude of its impact over the tweeting behavior of the users, $Mag(t, I)$.

### 3.4 Selection of Words Describing Events

Observing that clustering-based methods can in some cases lead to noisy event descriptions (Valkanas and Gunopulos, 2013), we adopt a different approach which we describe hereafter, with the aim to provide more semantically meaningful descriptions.

In order to limit information overload, we choose to bound the number of words used to describe an event. This bound is a fixed parameter noted $p$. We justify this choice because of the shortness of tweets. Indeed, because tweets contain very few words, it doesn't seem reasonable for an event to be associated with too many words (Weng and Lee, 2011).

**Identification of the candidate words** The set of candidate words for describing an event is the set of the words with the $p$ highest co-occurrence counts with the main word $t$ during the period of time $I$. The most relevant words are selected amongst the candidates based on the similarity between their temporal dynamics and the dynamics of the main word during $I$. For that, we compute a weight $w_q$ for each candidate word $t'_q$. We propose to estimate this weight from the time-series for $N_t^i$ and $N_{t'_q}^i$ with the correlation coefficient proposed by Erdem et al (2012). This coefficient, primarily designed to analyze stock prices, has two desirable properties for our application: (i) it is parameter-free and (ii) there is no stationarity assumption for the validity of this coefficient, contrary to common coefficients, *e.g.* Pearson's coefficient. This coefficient takes into account the lag difference of data points in order to better capture the direction of the co-variation of the two time-series over time. For the sake of conciseness, we directly give the formula for the approximation of the coefficient, given words $t$, $t'_q$ and the period of time $I = [a; b]$:

$$\rho_{Ot,t'_q} = \frac{\sum_{i=a+1}^{b} A_{t,t'_q}}{(b-a-1)A_t A_{t'_q}}$$

$$\text{where } A_{t,t'_q} = (N_t^i - N_t^{i-1})(N_{t'_q}^i - N_{t'_q}^{i-1}),$$

$$A_t^2 = \frac{\sum_{i=a+1}^{b}(N_t^i - N_t^{i-1})^2}{b-a-1} \text{ and}$$

$$A_{t'_q}^2 = \frac{\sum_{i=a+1}^{b}(N_{t'_q}^i - N_{t'_q}^{i-1})^2}{b-a-1}$$

This practically corresponds to the first order auto-correlation of the time-series for $N_t^i$ and $N_{t'_q}^i$. The proof that $\rho_O$ satisfies $|\rho_O| \leqslant 1$ using the Cauchy-Schwartz inequality is given by Erdem et al (2012). Eventually, we define the weight of the term $t'_q$ as an affine function of $\rho_O$ to conform with our definition of bursty topic, *i.e.* $0 \leqslant w_q \leqslant 1$:

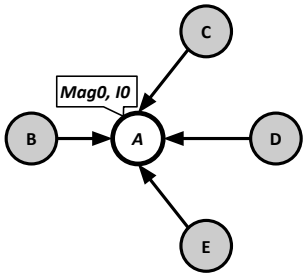
$$w_q = \frac{\rho_{Ot,t'_q} + 1}{2}$$

Because the temporal dynamics of very frequent words are less impacted by a particular event, this formulation – much like $tf \cdot idf$ – diminishes the weight of words that occur very frequently in the stream and increases the weight of words that occur less frequently, *i.e.* more specific words.

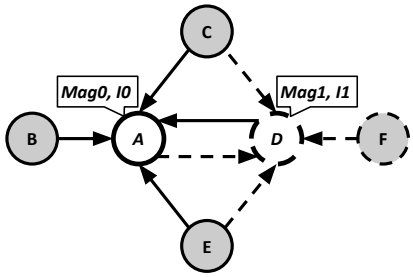
**Selection of the most relevant words** The final set of words retained to describe an event is the set $S$, such that $\forall t'_q \in S$, $w_q \geqslant \theta$. The parameters $p$ and $\theta$ allow the users of *MABED* to adjust the level of information and detail they require.

### 3.5 Generating the List of the Top k Events

Each time an event has been processed by the second component, it is passed to the third component. It is responsible for storing the description of the events while managing duplicated events. For that, it uses two graph structures: the event graph and the redundancy graph. The first is a directed, weighted, labeled graph that stores the descriptions of the detected events. The representation of an event $e$ in this graph is as follows. One node represents the main word $t$ and is labeled with the interval $I$ and the score *Mag*. Each related word $t'_q$ is represented by a node and has an arc toward the main word, which weight is $w_q$. The second structure is a




Storing the description of $e_0$ in the event graph.




Checking connectivity between events $e_0$ and $e_1$.

Adding a link between $A$ and $D$ in the redundancy graph.

**Fig. 3** Detecting duplication between events $e_0$ and $e_1$.

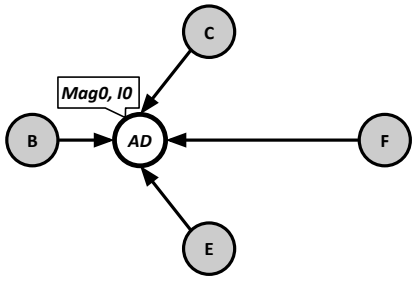



**Fig. 4** Merging event $e_0$ with duplicated event $e_1$.

simple undirected graph that is used to represent the relations between the eventual duplicated events, represented by their main words.

Let $e_1$ be the event that the component is processing. First, it checks whether it is a duplicate of an event that is already stored in the event graph or not. If it isn't the case, the event is added to the graph and the count of distinct events is incremented by one. Otherwise, assuming $e_1$ is a duplicate of the event $e_0$, a relation is added between $t_0$ and $t_1$ in the redundancy graph. When the count of distinct events reaches $k$, the duplicated events are merged and the list of the top $k$ most impactful events is returned. We describe how duplicated events are identified and how they are merged hereafter.

**Detecting duplicated events** The event $e_1$ is considered to be a duplicate of the event $e_0$ already stored in the event graph if (i) the main words $t_1$ and $t_0$ would be mutually connected and (ii) if the overlap coefficient between the periods of time $I_1$ and $I_0$ exceeds a fixed threshold. The overlap coefficient is defined as $\frac{|I_1 \cap I_0|}{\min(I_1, I_0)}$ and the threshold is noted $\sigma$, $\sigma \in ]0;1]$. In this case, the description of $e_1$ is stored aside and a relation is added between $t_1$ and $t_0$ in the redundancy graph. An example is shown in Figure 3, where $e_0 = \{A, \{B, C, D, E\}, I_0, Mag_0\}$, and $e_1 = \{D, \{C, E, F\}, I_1, Mag_1\}$.

**Merging duplicated events** Identifying which duplicated events should be merged together is equivalent to identifying the connected components in the redundancy graph. This is done in linear time, w.r.t to the numbers vertices and edges of the graph, using the algorithm described by Hopcroft and Tarjan (1973). In each connected component, there is exactly one node that corresponds to an event stored in the event graph. Its magnitude of impact and the related time interval remain the same, but its textual description is updated according to the following principle. The main word becomes the aggregation of the main words of all duplicated events. The words describing the updated event are the $p$ words among all the words describing the duplicated events with the $p$ highest weights. Figure 4 shows the description of the event resulting from the merging of the events $e_0$ and $e_1$, based on the event and redundancy graphs shown in Figure 3.

### 3.6 Overall algorithm

To conclude this section, Algorithm 1 sums-up the overall flow of *MABED*.

## 4 Experiments

In this section we present the main results of the extensive experimental study we conducted on both English and French Twitter data to evaluate *MABED*. In the quantitative evaluation, we demonstrate the relevance of the mention-anomaly-based approach and we quantify the performance of *MABED* by comparing it to state-of-the-art methods. To evaluate precision and recall, we ask human annotators to judge whether the detected events are meaningful and significant. In the qualitative evaluation, we show that the descriptions of the events detected by *MABED* are semantically and temporally more meaningful than the descriptions provided by existing methods, which favors an easy understanding of the results. Lastly, we analyse the detected

**Algorithm 1:** Overall algorithm for *MABED*.

**Data**: A corpus $\mathcal{C}$ of $N$ tweets partitioned into $n$ time-slices of equal length and the corresponding vocabularies, $V$ and $V_@$
**Parameters**: $k > 0$, $p > 0$, $\theta \in [0; 1]$ and $\sigma \in ]0; 1]$
**Result**: The ordered list $L$ of the $k$ most impactful detected events

```
/* First phase                                          */
```
Initialize the stack $P$ which stores the detected events during the first phase;
**for** *each word* $t \in V_@$ **do**
  Identify the interval $I = [a; b]$ such that : $Mag(t, I) = \max\{\sum_{i=a}^{b} anomalie(t, i)$;
  Add the event $e = [t, \emptyset, I, Mag(t, I)]$ in $P$;
**end**
Sort the event stack $P$ by descending magnitude of impact;
```
/* Second phase                                         */
```
Initialize the event graph $G_E$ and the redundancy graph $G_R$;
Set *count* to 0;
**while** *count* $< k$ *and* $|P| > 0$ **do**
  Pop the event $e$ on top of the stack $P$;
  Select the words that describe $e$, with parameters $p$ and $\theta$;
  **if** *$e$ is redundant with the event $e'$ which is already in $G_E$, for a given $\sigma$* **then**
    Add a link between the mains word of events $e$ and $e'$ in $G_R$;
    Store the description of $e$ aside;
  **else**
    Insert the description $e$ in graph $G_E$;
    Increment *count*;
  **end**
**end**
Identify which events should be merged based on $G_R$ then update $G_E$;
Transform graph $G_E$ into list $L$;
Sort $L$ ;
**return** $L$;

event with regard to user communities and find that *MABED* can help understanding their interests.

### 4.1 Experimental Setup

**Corpora** Since the Twitter corpora used in prior work aren't available we base our experiments on two different corpora. The first corpus – noted $\mathcal{C}_{en}$ – contains 1,437,126 tweets written in English, collected with a user-centric strategy. They correspond to all the tweets published in November 2009 by 52,494 U.S.-based users (Yang and Leskovec, 2011). This corpus contains a lot of noise and chatter. According to the study conducted by PearAnalytics (2009), the proportion of non-event-related tweets could be as high as 50%. The second corpus – noted $\mathcal{C}_{fr}$ – contains 2,086,136 tweets written in French, collected with a keyword-based strategy. We have collected these tweets in March 2012, during the campaign for the 2012 French presidential elections, using the Twitter streaming API with a query consisting of the names of the main candidates running for president. This corpus is focused on French politics. Trivial words are removed from both corpora based on English and French standard stop-word lists. All timestamps are in UTC. Table 2 gives further details about each corpus.

**Table 2** Corpus Statistics. @: proportion of tweets that contain mentions, *RT*: proportion of retweets.

| Corpus | Tweets | Authors | @ | *RT* |
|---|---|---|---|---|
| $\mathcal{C}_{en}$ | 1,437,126 | 52,494 | 0.54 | 0.17 |
| $\mathcal{C}_{fr}$ | 2,086,136 | 150,209 | 0.68 | 0.43 |

**Baselines for comparison** We consider two recent methods from the literature: *ET* (clustering-based) and *TS* (term-weighting-based). *ET* is based on the hierarchical clustering of bigrams using content and appearance patterns similarity (Parikh and Karlapalem, 2013). *TS* is a normalized frequency metric for identifying n-grams that are related to events (Benhardus and Kalita, 2013). We apply it to both bigrams (*TS2*) and trigrams (*TS3*). We also consider a variant of *MABED*, noted *α-MABED*, that ignores the presence of mentions in tweets. This means that the first component detects events and estimates their magnitude of impact based on the values of $N_t^i$ instead of $N_{@t}^i$. The reasoning for excluding a comparison against topic-modeling-based methods is that in preliminary experiments we found that they performed poorly and their computation times were prohibitive.

**Parameter setting** For *MABED* and *α-MABED*, we partition both corpora using 30 minute time-slices, which allows for a good temporal precision while keeping the number of tweets in each time-slice large enough. The maximum number of words describing each event, $p$, and the weight threshold for selecting relevant words, $\theta$, are parameters that allow the user to define the required level of detail. Given that the average number of words per sentence on Twitter is 10.7 according to the study conducted by Oxford (2009), we fix $p$ to 10. For the purpose of the evaluation, we set $\theta = 0.7$ so judges are only presented with words that are closely related to each event. There is a parameter that can affect the performance of *MABED*: $\sigma$. In the following, we report results for $\sigma = 0.5$ (we discuss the impact of $\sigma$ in Section 4.1).

For *ET* and *TS*, because they assume a fixed duration for all events – which corresponds to the length of one time-slice – we partition both corpora using 1-day time-slices like in prior work. *ET* has two parameters, for which we use optimal values provided by the authors.

**Evaluation metrics** The corpora don't come with ground truth, therefore we have asked two human annotators to judge whether the detected events are meaningful and significant. The annotators are French graduate students who aren't involved in this project. Their task consisted in reading the descriptions of the events detected by each method, and independently assign a rate to each description. This rate can be either 1, if the annotator decides that the description is meaningful and related to a significant event (*i.e.* an event that may be covered in traditional media), or 0 in any other cases. They also had to identify descriptions similar to ones which had previously been rated, in order to keep track of duplicates. Considering that annotating events is a time consuming task for the annotators, we limit the evaluation to the 40 most impactful events detected by each method (*i.e.* $k = 40$) in each corpus. We measure precision as the ratio of the number of detected events that both annotators have rated 1, which we refer to as $k'$, to the total number of detected events, $k$:

$$P = \frac{k'}{k}$$

Based on the number of duplicated events, $k''$, we define recall as the fraction of distinct significant events among all the detected events (Li et al, 2012):

$$R = \frac{k' - k''}{k}$$

We also measure the DERate (Li et al, 2012), which denotes the percentage of events that are duplicates among all the significant events detected, that is to say DERate $= k''/k'$.

## 4.2 Quantitative Evaluation

Hereafter, we discuss the performance of the five considered methods, based on the rates assigned by the annotators. The inter-annotator agreement, measured with Cohen's Kappa (Landis and Koch, 1977), is $\kappa \simeq 0.76$, showing a strong agreement. Table 3 (page 10) reports the precision, the F-measure defined as the harmonic mean of precision and recall (*i.e.* $2 \cdot \frac{P \cdot R}{P+R}$), the DERate and the running-time of each method for both corpora.

**Comparison against baselines** We notice that *MABED* achieves the best performance on the two corpora, with a precision of 0.775 and F-measure of 0.682 on $\mathcal{C}_{en}$, and a precision and a F-measure of 0.825 on $\mathcal{C}_{fr}$. Although *ET* yields a better *DERate* on $\mathcal{C}_{en}$, it still achieves lower precision and recall than *MABED* on both corpora. Furthermore, we measure an average relative gain of 17.2% over $\alpha$-*MABED* in the F-measure, which suggests that considering the mentioning behavior of users leads to more accurate detection of significant events in Twitter. Interestingly, we notice that *MABED* outperforms all baselines in the F-measure with a bigger margin on $\mathcal{C}_{en}$, which contains a lot more noise than $\mathcal{C}_{fr}$ – with up to 50% non-event-related tweets according to the study conducted by PearAnalytics (2009). This suggests that considering the mentioning behavior of users also leads to more robust detection of events from noisy Twitter content. The DERate reveals that none of the significant events detected in $\mathcal{C}_{fr}$ by *MABED* were duplicated, whereas 6 of the significant events detected in $\mathcal{C}_{en}$ are duplicates. Furthermore, we find that the set of events detected by the four baseline methods is a sub-set of the events detected by *MABED*. Further analysis of the results produced by $\alpha$-*MABED*, *TS2* and *TS3* reveals that most of non-significant events they detected are related to spam. The fact that most of these irrelevant events aren't detected by *MABED* suggest that considering the presence of mentions in tweets helps filtering away spam. Concerning *ET*, the average event description is 17.25 bigrams long (*i.e.* more than 30 words). As a consequence, the descriptions contain some unrelated words. Specifically, irrelevant events are mostly sets of unrelated words that don't make any sense. This is due in part to the fact that clustering-based approaches are prone to aggressively grouping terms together, as Valkanas and Gunopulos (2013) stated in a previous study.

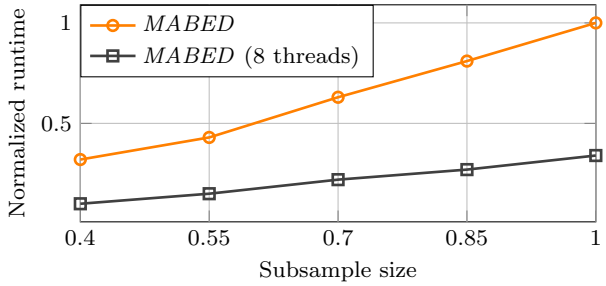



**Fig. 5** Runtime comparison versus subsample size.

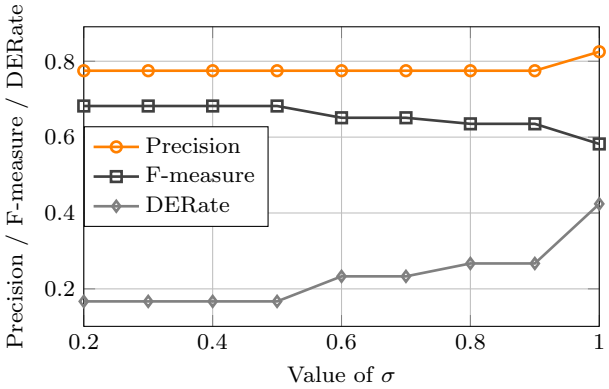



**Fig. 6** Precision, F-measure and DERate of *MABED* on $\mathcal{C}_{en}$ for different values of $\sigma$.

**Table 3** Performance of the five methods on the two corpora.

| | Corpus: $\mathcal{C}_{en}$ | | | | Corpus: $\mathcal{C}_{fr}$ | | | |
|---|---|---|---|---|---|---|---|---|
| Method | Precision | F-measure | DERate | Running-time | Precision | F-measure | DERate | Running-time |
| *MABED* | 0.775 | 0.682 | 0.167 | 96s | 0.825 | 0.825 | 0 | 88s |
| *α-MABED* | 0.625 | 0.571 | 0.160 | 126s | 0.725 | 0.712 | 0.025 | 113s |
| *ET* | 0.575 | 0.575 | 0 | 3480s | 0.700 | 0.674 | 0.071 | 4620s |
| *TS2* | 0.600 | 0.514 | 0.250 | 80s | 0.725 | 0.671 | 0.138 | 69s |
| *TS3* | 0.375 | 0.281 | 0.4 | 82s | 0.700 | 0.616 | 0.214 | 74s |

**Efficiency** It appears that *MABED* and *TS* have running-times of the same order, whereas *ET* is orders of magnitude slower, which is due to the clustering step that requires computing temporal and semantical similarity between all bigrams. We also observe that *MABED* runs faster than *α-MABED*. The main reason for this is that $|V_@| \leqslant |V|$, which speeds up the first phase. It should be noted that the running-times given in Table 3 don't include the time required for preparing vocabularies and pre-computing term frequencies, which is more important for methods that rely on bigrams or trigrams. We evaluate the scalability of (i) *MABED* and (ii) a parallelized version of *MABED* (8 threads), by measuring their running-times on random subsamples of both corpora, for subsample sizes varying from 40% to 100%. Figure 5 shows the average normalized running--time versus subsample size. This means that the runtimes measured on a corpus are normalized by the longest runtime on this corpus and are then averaged for *MABED* and *MABED* (8 threads). We notice that runtimes grow linearly in size of the subsample. Furthermore, we note that *MABED* (8 threads) is on average 67% faster than *MABED*.

**Impact of $\sigma$ on *MABED*** While the list of events is constructed by *MABED*, the overlap threshold $\sigma$ controls the sensitivity to duplicated events. Figure 6 plots the precision, F-measure and DERate of *MABED* on $\mathcal{C}_{en}$ for values of $\sigma$ ranging from 0.2 to 1. We observe that the value of $\sigma$ mainly impacts the DERate. More specifically, the DERate increases along the increase of $\sigma$ as fewer duplicated events are merged. For $\sigma = 1$, the precision increases to 0.825 because of the high percentage of duplicated significant events. Globally, it appears that the highest F-measure is attained for values of $\sigma$ ranging from 0.2 to 0.5. However, even using $\sigma = 1$, *MABED* achieves a F-measure of 0.582, which is higher than all baselines on $\mathcal{C}_{en}$.

## 4.3 Qualitative Evaluation

Next, we qualitatively analyze the results of *MABED* and show how they provide relevant information about the detected events. Table 4 (page 11) lists the top 25 events with highest magnitude of impact over the crowd in $\mathcal{C}_{en}$. From this table, we make several observations along three axes: readability, temporal precision and redundancy.

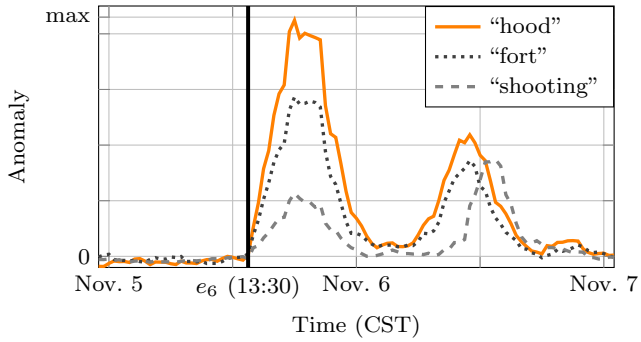



**Fig. 7** Measured anomaly for the words "hood", "fort" and "shooting" between Nov. 5 and Nov. 7 midnight (CST).

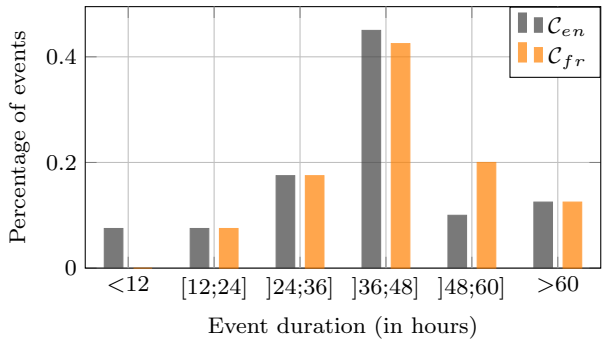



**Fig. 8** Distribution of the duration of the events detected with *MABED*.

**Readability** We argue that highlighting main words allows for an easy reading of the description, more especially as main words often correspond to named entities, *e.g.* Fort Hood ($e_6$), Chrome ($e_7$), Tiger Woods ($e_8$), Obama ($e_{13}$). This favors a quicker understanding of events by putting into light the key places, products or actors at the heart of the events, in contrast with existing methods that identify bags of words or n-grams. What is more, *MABED* ranks the words that describe each event and limits their number, which again favors the interpretation of events.

**Temporal precision** *MABED* dynamically estimates the period of time during which each event is discussed on Twitter. This improves the temporal precision as

**Table 4** Top 25 events with highest magnitude of impact over the crowd, detected by *MABED* in $\mathcal{C}_{en}$. Main words are in bold and time intervals are given in UTC (Coordinated Universal Time).

| $e_{\#}$ | Time interval | Topic |
|---|---|---|
| 1 | from 25 09:30 to 28 06:30 | **thanksgiving, turkey**: hope (0.72), happy (0.71)<br>*Twitter users celebrated Thanksgiving* |
| 2 | from 25 09:30 to 27 09:00 | **thankful**: happy (0.77), thanksgiving (0.71)<br>*Related to event #1* |
| 3 | from 10 16:00 to 12 08:00 | **veterans**: served (0.80), country (0.78), military (0.73), happy (0.72)<br>*Twitter users celebrated the Veterans Day that honors people who have served in the U.S. army* |
| 4 | from 26 13:00 to 28 10:30 | **black**: friday (0.95), amazon (0.75)<br>*Twitter users were talking about the deals offered by Amazon the day before the "Black Friday"* |
| 5 | from 07 13:30 to 09 04:30 | **hcr, bill, health, house, vote**: reform (0.92), passed (0.91), passes (0.88)<br>*The House of Representatives passed the health care reform bill on November 7* |
| 6 | from 05 19:30 to 08 09:00 | **hood, fort**: ft (0.92), shooting (0.83), news (0.78), army (0.75), forthood (0.73)<br>*The Fort Hood shooting was a mass murder that took place in a U.S. military post on November 5* |
| 7 | from 19 04:30 to 21 02:30 | **chrome**: os (0.95), google (0.87), desktop (0.71)<br>*On November 19, Google released Chrome OS's source code for desktop PC* |
| 8 | from 27 18:00 to 29 05:00 | **tiger, woods**: accident (0.91), car (0.88), crash (0.88), injured (0.80), seriously (0.80)<br>*Tiger Woods was injured in a car accident on November 27, 2009* |
| 9 | from 28 22:30 to 30 23:30 | **tweetie, 2.1, app**: retweets (0.93), store (0.90), native (0.89), geotagging (0.88)<br>*The iPhone app named Tweetie (v2.1), hit the app store with additions like retweets and geotagging* |
| 10 | from 29 17:00 to 30 23:30 | **monday, cyber**: deals (0.84), pro (0.75)<br>*Twitter users were talking about the deals offered by online shops for the "Cyber Monday"* |
| 11 | from 10 01:00 to 12 03:00 | **linkedin**: synced (0.86), updates (0.84), status (0.83), twitter (0.71)<br>*Starting from November 10, LinkedIn status updates can be synced with Twitter* |
| 12 | from 04 17:00 to 06 05:30 | **yankees, series**: win (0.84), won (0.84), fans (0.78), phillies (0.73), york (0.72)<br>*The Yankees baseball team defeated the Phillies to win their 27th World Series on November 4* |
| 13 | from 15 09:00 to 17 23:30 | **obama**: chinese (0.75), barack (0.72), twitter (0.72), china (0.70)<br>*Barack Obama admitted that he'd never used Twitter but Chinese should be able to* |
| 14 | from 25 10:00 to 26 10:00 | **holiday**: shopping (0.72)<br>*Twitter users started talking about the "Black Friday", a shopping day and holiday in some states* |
| 15 | from 19 21:30 to 21 16:00 | **oprah, end**: talk (0.81), show (0.79), 2011 (0.73), winfrey (0.71)<br>*On November 19, Oprah Winfrey announced her talk show will end in September 2011* |
| 16 | from 07 11:30 to 09 05:00 | **healthcare, reform**: house (0.91), bill (0.88), passes (0.83), vote (0.83), passed (0.82)<br>*Related to event #5* |
| 17 | from 11 03:30 to 13 08:30 | **facebook**: app (0.74), twitter (0.73)<br>*No clear corresponding event* |
| 18 | from 18 14:00 to 21 03:00 | **whats**: happening (0.76), twitter (0.73)<br>*Twitter started asking "What's happening?" instead of "What are you doing?" from November 18* |
| 19 | from 20 10:00 to 22 00:00 | **cern**: lhc (0.86), beam (0.79)<br>*On November 20, proton beams were successfully circulated in the ring of the LHC (CERN)* |
| 20 | from 26 08:00 to 26 15:30 | **icom**: lisbon (0.99), roundtable (0.98), national (0.88)<br>*The I-COM roundtable about market issues in Portugal took place on November 26* |
| 21 | from 03 23:00 to 05 10:00 | **maine**: voters (0.76), marriage (0.71)<br>*On November 4, Maine voters repealed a state law granting same-sex couples the right to marry* |
| 22 | from 07 13:00 to 10 16:30 | **droid**: verizon (0.75), iphone (0.72), video (0.70)<br>*On November 7, Verizon stores released the new DROID phone, promoted as an iPhone alternative* |
| 23 | from 18 14:00 to 20 09:00 | **read**: blog (0.76), article (0.74)<br>*No clear corresponding event* |
| 24 | from 02 05:00 to 03 19:00 | **wave**: guide (0.81), google (0.73)<br>*The complete Google Wave guide was released on November 9* |
| 25 | from 18 10:30 to 20 09:00 | **talk, show**: oprah (0.89), 2011 (0.85), end (0.77)<br>*Related to event # 15* |

compared to existing methods, that typically report events on a daily basis. We illustrate how this improves the quality of the results with the following example. The 6$^{\text{th}}$ event corresponds to Twitter users reporting the Fort Hood shooting that, according to Wikipedia[1], happened on November 5, 2009 between 13:34 and 13:44 CST (*i.e.* 19:34 and 19:44 UTC). The burst of activity engendered by this event is first detected by *MABED* in the time-slice covering the 19:30-20:00 UTC period. *MABED* gives the following description:

[1]Source: `http://en.wikipedia.org/wiki/Fort_Hood_shooting`

**(i)** *11-05 19:30 to 11-08 9:00*; **(ii)** *hood, fort*; **(iii)** *ft (0.92), shooting (0.83), news (0.78), army (0.75), forthood (0.73).*

We can clearly understand that (i) something happened around 7:30pm UTC, (ii) at the Hood Fort and that (iii) it is a shooting. In contrast, *α-MABED* fails at detecting this event on November 5 but reports it on November 7 when the media coverage was the highest.

**Redundancy** Some events have several main words, *e.g.* events $e_1$, $e_5$, $e_6$, $e_8$. This is due to merges operated by the third component of *MABED* to avoid duplicated events. Redundancy is further limited because of the dynamic estimation of each event duration. We may continue using event $e_6$ to illustrate that. Figure 7 plots the evolution of the anomaly measured for the words "hood", "fort" and "shooting" between November 5 and November 7. We see that the measured anomaly is closer to 0 during the night (local time), giving a "dual-peak" shape to the curves. Nevertheless, *MABED* reports a unique event which is discussed for several days, instead of reporting distinct consecutive 1-day events. The importance of dynamically estimating the duration of events is further illustrated by Figure 8, which shows the distributions of event duration for both corpora. It reveals that some events are discussed during less than 12 hours whereas some are discussed for more than 60 hours. We note that event durations in $\mathcal{C}_{fr}$ are normally distributed and that these politics-related events tend to be discussed for a longer duration than the events detected in $\mathcal{C}_{en}$. This is consistent with the empirical study presented by Romero et al (2011), which states that controversial and more particularly political topics are more persistent than other topics on Twitter.

### 4.4 Analysis of Detected Events

In this section we analyze events detected by *MABED* in $\mathcal{C}_{en}$, with regard to the communities detected in the structure of the network that interconnects the authors of the tweets in that corpus. Our goal is to show that *MABED* can help understanding users', or user communities' interests.

**Network structure** The 52,494 authors of the tweets in $\mathcal{C}_{en}$ are interconnected by 5,793,961 following relationships (Kwak et al, 2010). This forms a directed, connected network, which diameter is 8, with an average path length of 2.55 and a clustering coefficient of 0.246. We measure a small-world-ness metric (Humphries et al, 2006) of $s = 47.2$, which means – according to the definition given by Humphries et al (2006), *i.e.* $s > 1$ – this network is a small-world network.

**Detecting communities in the network** In order to find communities in this network, we apply the Louvain method proposed by Blondel et al (2008), which has been notably used to detect communities in online social networks in several studies (Haynes and Perisic, 2010; Kim et al, 2013). It is a heuristic-based, greedy method for optimizing modularity (Newman, 2006). It detects two communities: $c_0$, comprised of 25,625 users, and $c_1$, comprised of 26,869 users. By $\mathcal{C}_{en}(c_0)$ and $\mathcal{C}_{en}(c_1)$, we denote, respectively, the corpus of 479,899 tweets published by the users belonging to $c_0$ and the corpus of 932,699 tweets published by users belonging to $c_1$.

**Characterizing communities' interests** We extract the list $L_0$ (respectively $L_1$) of the $k$ most impactful events from the corpus $\mathcal{C}_{en}(c_0)$ (respectively $\mathcal{C}_{en}(c_1)$). Here we choose to fix $k = 10$, so only events that engendered a significant amount of reactions in the related community are considered. In order to compare the two communities' interests, we first manually label each detected event with one the following categories (McMinn et al, 2013):

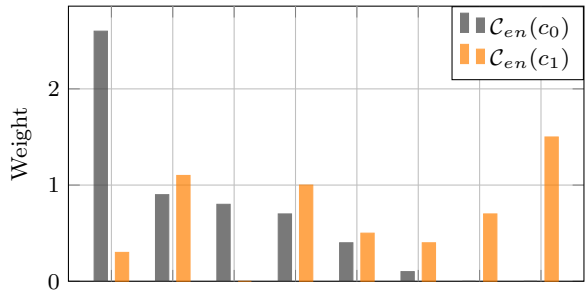
- Armed Conflicts and Attacks;
- Sports;
- Disasters and Accidents;
- Art Culture and Entertainment;
- Business and Economy;
- Law Politics and Scandals;
- Science and Technology;
- Miscellaneous.

Then, we measure the weight of each category in each event list and compute the category weight distribution for each community. The weight of a given category is obtained by the following formula:

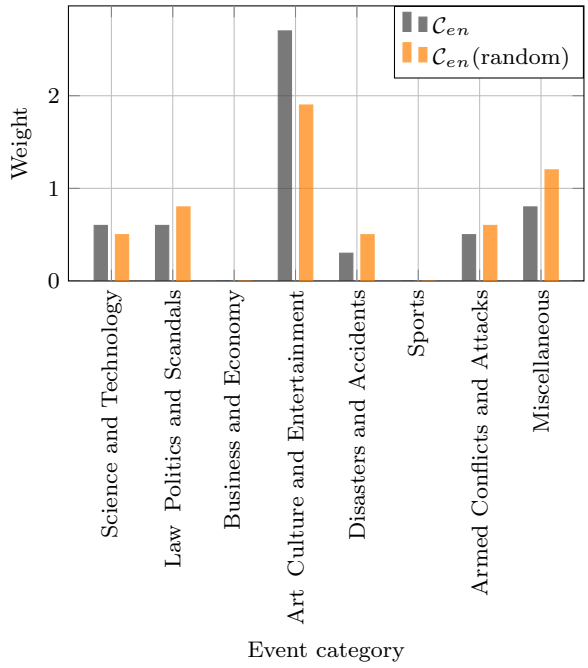
$$\text{Weight}_{\text{category}} = \sum_{e \in E} 1 - ((\text{rank}(e) - 1) \times 0.1)$$

where $E$ is the set of events labelled with this category. Thus, the contribution of an event to the weight of the related category linearly diminishes with its rank in the list, *e.g.* an event ranked $1^{\text{st}}$ contributes for a weight of 1, whereas an event ranked $10^{\text{th}}$ contributes for a weight of 0.1.

For comparison purpose, we do the same for events detected in $\mathcal{C}_{en}$ and $\mathcal{C}_{en}$(random). The $\mathcal{C}_{en}$(random) corpus contains 725,806 tweets that corresponds to the subset of tweets published by 26,000 randomly chosen authors (*i.e.* about the number of users in communities $c_0$ or $c_1$) in $\mathcal{C}_{en}$. Figure 9(a) shows the weight distribution for communities $c_0$ and $c_1$. We note that, visually, the two distributions seem quite different. For instance, we notice that the "Miscellaneous" category has the highest weight for $c_1$ whereas none of the events




(a) Events detected in $\mathcal{C}_{en}(c_0)$ and $\mathcal{C}_{en}(c_1)$.




(b) Events detected in $\mathcal{C}_{en}$ and $\mathcal{C}_{en}$(random).

**Fig. 9** Category weight distribution for events detected with *MABED* in $\mathcal{C}_{en}(c_0)$, $\mathcal{C}_{en}(c_1)$, $\mathcal{C}_{en}$ and $\mathcal{C}_{en}$(random).

detected in $\mathcal{C}_{en}(c_0)$ belongs to that category. On the opposite, the "Science and Technology" category has the highest weight for $c_0$ whereas it has the second lowest weight for $c_1$. We measure a negative linear correlation of -0.36 between the two distributions using Pearson's coefficient, which reinforce this assessment. Figure 9(b) shows the distribution of event category weight for $\mathcal{C}_{en}$ and $\mathcal{C}_{en}$(random). In this case, we measure a linear correlation of 0.93, which means that the two distributions are very similar. These results indicate that the communities detected in the network structure are also relevant in terms of users' interests. These results also shed light on the interplay between the social structure, *i.e.* who follows whom, and the topical structure, *i.e.* who's interested in what, in Twitter. More specifically, they complement the findings from Romero et al (2013) – who have found a relationship between the hashtags users adopt and their social ties – and suggest that the social network structure can influence event detection. From a different perspective, these results show that event detection can help understanding user communities' interests.

## 5 Implementation and Visualizations

We provide a parallel implementation[2,3] of *MABED*. It is also included in *SONDY* (Guille et al, 2013), an open-source social media data mining software that implements several state-of-the-art methods for event detection in social media. To ensure an efficient exploration of the events detected by *MABED*, we also develop three visualizations, which we describe below.

**Time-oriented visualization** It is based on an interactive timeline that allows the user to explore the detected events through time. As an example, Figure 10 shows the time-oriented visualization generated from the events detected in $\mathcal{C}_{en}$. As one can see, the timeline is divided into two parts. The lower part is a ribbon labelled with events in chronological order (fig. 10.1). Selecting a label (*i.e.* the main term of an event) in the lower part updates the upper part of the visualization with details about the related event. More specifically, the upper part displays the temporal and textual descriptions extracted with *MABED* (fig. 10.2.a), an image (fig. 10.2.b) and a hypertext (fig. 10.2.c). In the current implementation, these correspond to the top image and the description of the top page returned by the Bing search engine, using the description extracted by *MABED* as a query.

This visualization provides a chronological overview of the events detected in a tweet corpus. In addition, the hypertexts offer quick access to resources that can help learning more about these events. For instance, the hypertext associated to the event selected on the timeline depicted in Figure 10, reveals that, on November 19, 2009, Google released Chrome OS's source code and demonstrated an early version of this operating system for desktop computers.

**Impact-oriented visualization** It is an interactive chart that allows analyzing the magnitude of impact of the detected events. More precisely, it plots the mention-anomaly function related to each event. Figure 11 depicts this visualization for the top 20 events detected in $\mathcal{C}_{en}$. The interactive legend (fig.11.1) can be used to display (fig.11.2) only the functions which are of interest by clicking on them. Single clicking on a legend item removes the corresponding function from

[2]Binaries: `http://mediamining.univ-lyon2.fr/mabed`
[3]Sources: `https://github.com/AdrienGuille/MABED`

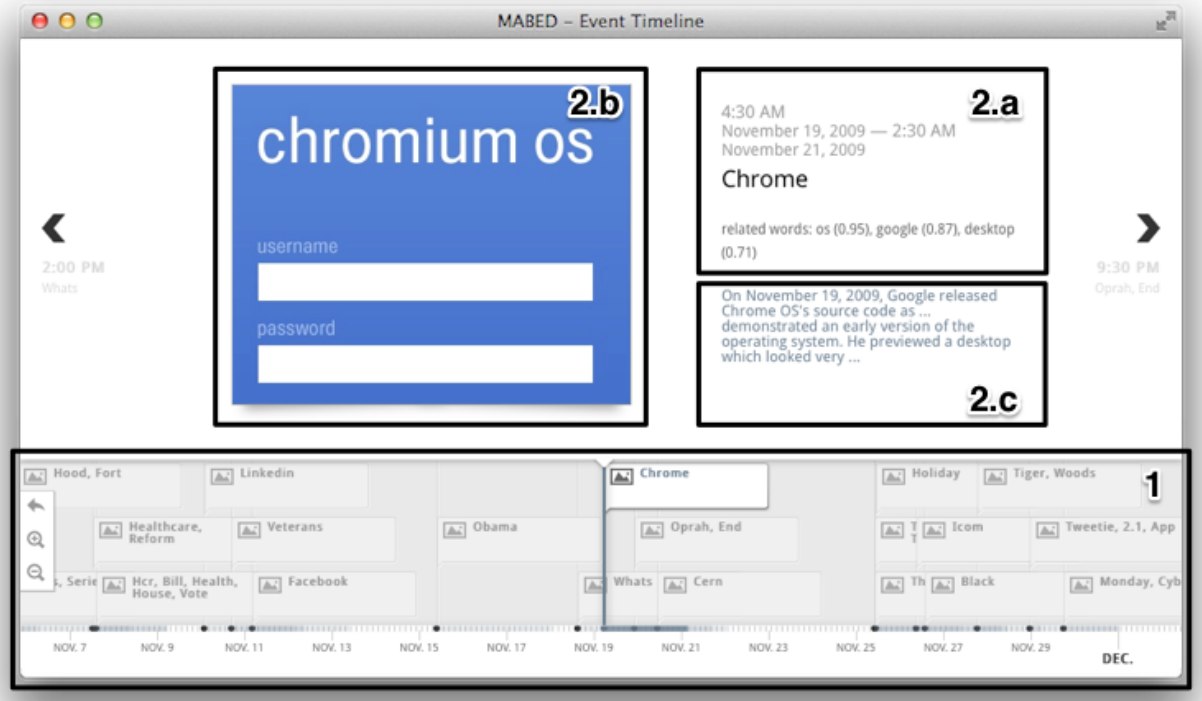



**Fig. 10** The time-oriented visualization. The lower part is a ribbon (1) labeled with events in chronological order. The upper part displays details about the event selected in the lower part: the description extracted with *MABED* (2.a), an image (2.b) and a hypertext (2.c).

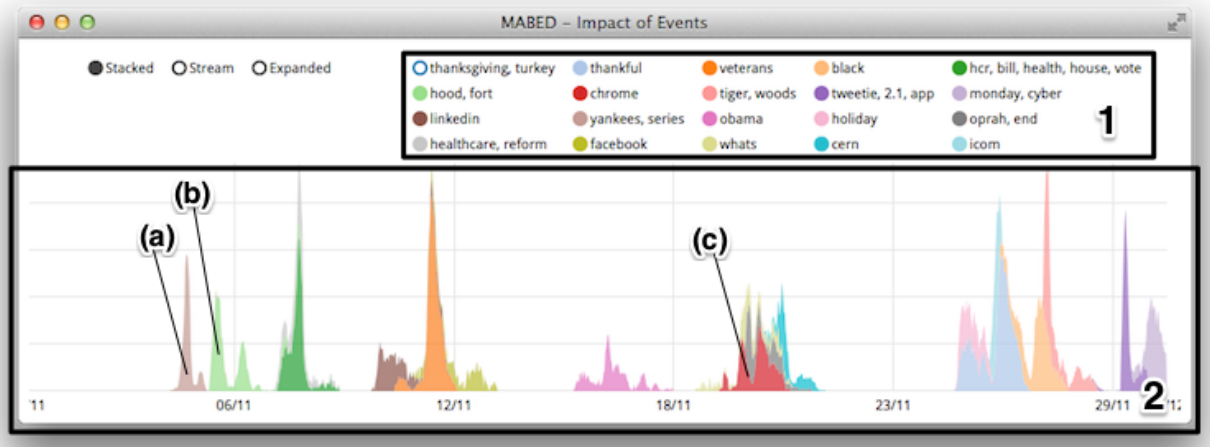



**Fig. 11** The impact-oriented visualization is a chart (1) that plots the magnitude of impact of the detected events (2). Each event is associated to a different color.

the chart while double clicking on a item makes it the only visible one in the chart.

This visualization helps analyzing the temporal patterns that describe how Twitter users reacted to the detected events. For instance, we observe that different events trigger different patterns: some events engender a single significant peak of reactions (*e.g.* fig.11.a), some events generate successive peaks of decreasing strength (*e.g.* fig.11.b), while other events engender successive increasing peaks of attention (*e.g.* fig.11.c).

**Topic-oriented visualization** It is based on the event graph constructed by *MABED* during the second phase. Figure 12 shows the event graph constructed from $\mathcal{C}_{fr}$. Main terms are represented with grey nodes (fig. 12.1), whose diameter is proportional to the magnitude of impact of the corresponding event. Related words are represented by blue nodes (fig. 12.3), which are connected to main terms by edges (fig. 12.2) whose thickness is proportional to the related weight. For the sake of readability, nodes' labels are hidden by default, but the user can click a grey node in order to reveal the main term and related words describing the event.

This visualization helps identifying similar events by topic. It also helps discovering words which are common to several events. This could be useful in cases when one wants to quickly identify events involving, *e.g.* a specific actor or place. For instance, we spot two nodes (fig.12.a, b) which describe many events. They correspond to the two main candidates for the 2012 presidential elections. Interestingly, even though they appear in many different events, they appear together in only one single event.

### 5.1 Case study: Monitoring the French Political Conversation on Twitter

**Setup** *MABED* has been used from December 2013 until November 2014 to continuously analyze the French political conversation on Twitter. For this purpose, it has been coupled with a system fetching tweets in real-time about the President of the French Republic, Francois Hollande, via the Twitter streaming API. During this period, the highest crawling rate we have reached was about 150,000 tweets in 24 hours. Every 10 minutes, the visualizations were refreshed with the five most impactful events detected from the tweets received in the last 24 hours (partioned into 144 time-slices of 10 minutes each).

**Results** It has helped us to understand the political conversation on Twitter, and it has revealed interesting clues about how public opinion evolves on Twitter. The visualizations, most notably the impact-oriented visualizations in this case, have also shed light on specific patterns. As an example, Figure 13 shows two impact-oriented visualizations. They were automatically generated from the five most impactful events detected in two successive 24-hour periods: (i) December 13$^{\text{th}}$, 2013 and (ii) December 14$^{\text{th}}$, 2013. We notice that the two distinct sets of detected events are distributed differently throughout each day. On December 13$^{\text{th}}$, we observe that Twitter users focus their attention on various successive events throughout the day. On the other hand, on the 14$^{\text{th}}$ we observe an opposite pattern since all the events are discussed simultaneously during the second half of the day. Further analysis, indicates that all five events detected on December 13$^{\text{th}}$ are closely related. In contrast, the events detected on the 14$^{\text{th}}$ are loosely related. One possible explanation for the pattern we observe on December 13$^{\text{th}}$ is that related events can "compete" or "collaborate". The mention-anomaly measured for competing events should thus be negatively correlated, while the mention-anomaly measured for cooperating events should be positively correlated. For instance, we observe that events $e3$ and $e4$ take place simultaneously and are likely to collaborate. Eventually, both of them end when events $e2$ and $e1$ (likely to be competing with $e3$ and $e4$) start.

## 6 Conclusion

In this paper, we developed *MABED*, a mention-anomaly-based method for event detection in Twitter. In contrast with prior work, *MABED* takes the social aspect of tweets into account by leveraging the creation frequency of mentions that users insert in tweets to engage discussion. Our approach also differ from prior work in that it dynamically estimates the period of time during which each event is discussed on Twitter. The experiments we conducted have shown that *MABED* has a linear runtime in the corpus size. They have also demonstrated the relevance of our approach. Quantitatively speaking, *MABED* yielded better performance in all our tests than $\alpha$-*MABED* – a variant that ignores mentions – and also outperformed two recent methods from the literature. Qualitatively speaking, we have shown that the highlighting of main words improves the readability of the descriptions of events. We have also shown that the temporal information provided by *MABED* is very helpful. On the one hand, it clearly indicates when real-world events happened. On the other hand, dynamically identifying the period of time during which each event is discussed limits the fragmentation of events. By analyzing the detected events with regard to the user communities detected in the social network structure, we have shed light on the interplay between social and topical structure in Twitter. In particular, we

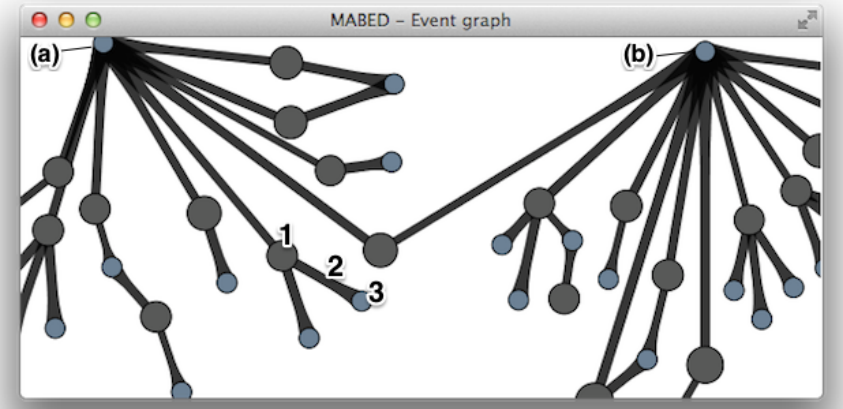



**Fig. 12** The topic-oriented visualization is an interactive drawing of the event graph. Grey nodes (1) correspond to main terms while blue nodes (3) correspond to related words. Edges (2) connect words that describe the same event.

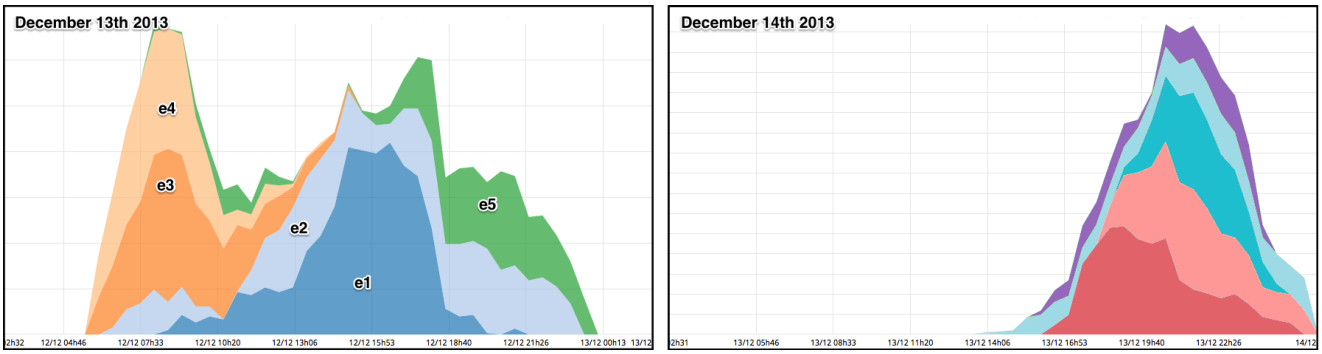



**Fig. 13** On the left: the impact-oriented visualization generated based on the five most impactful events detected from tweets collected on December 13$^{th}$, 2013. On the right: the impact-oriented visualization generated based on the five most impactful events detected from tweets collected on December 14$^{th}$, 2013. Each event is associated to a different color.

have found that *MABED* can help understanding user communities' interests. Moreover, we presented three visualizations designed to help with the exploration of the detected events. Finally, we described how we leveraged *MABED* and the visualizations we developed in order to continuously monitor and analyze the French political conversation on Twitter from December 2013 until November 2014.

As part of future work, we plan to investigate the effectiveness of utilizing more features to model the discussions between users (*e.g.* number of distinct users, users' geolocations). Another interesting direction for future work is to incorporate sentiment analysis in the event detection process to further enrich event descriptions.

**Acknowledgements** The authors would like to thank Michael Gauthier for helpful suggestions. This work was supported in part by the French National Research Agency and the ImagiWeb project (ANR-2012-CORD-002-01).

## References

Aggarwal CC, Subbian K (2012) Event detection in social streams. In: SDM, pp 624–635

Aiello LM, Petkos G, Martin C, Corney D, Papadopoulos S, Skraba R, Goker A, Kompatsiaris Y, Jaimes A (2013) Sensing trending topics in twitter. IEEE Trans Multimedia 15(6):1–15

AlSumait L, Barbará D, Domeniconi C (2008) On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In: ICDM, pp 3–12

Becker H, Iter D, Naaman M, Gravano L (2011) Beyond trending topics: Real-world event identification

on twitter. In: ICWSM, pp 438–441

Benhardus J, Kalita J (2013) Streaming trend detection in twitter. IJWBC 9(1):122–139

Bentley J (1984) Programming pearls: algorithm design techniques. CACM 27(9):865–873

Bernstein MS, Suh B, Hong L, Chen J, Kairam S, Chi EH (2010) Eddi: Interactive topic-based browsing of social status streams. In: UIST '10, pp 303–312

Blei D, Ng A, Jordan M (2003) Latent dirichlet allocation. JMLR 3:993–1022

Blondel V, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment P10008:1–12

Erdem O, Ceyhan E, Varli Y (2012) A new correlation coefficient for bivariate time-series data. In: MAF, pp 58–73

Fan TH, Lee S, Lu HI, Tsou TS, Wang TC, Yao A (2003) An optimal algorithm for maximum-sum segment and its application in bioinformatics. In: CIAA, pp 251–257

Fung GPC, Yu JX, Yu PS, Lu H (2005) Parameter free bursty events detection in text streams. In: VLDB, pp 181–192

Guille A, Favre C, Hacid H, Zighed D (2013) Sondy: An open source platform for social dynamics mining and analysis. In: SIGMOD, pp 1005–1008

Haynes J, Perisic I (2010) Mapping search relevance to social networks. In: SNA-KDD, pp 11–20

Hopcroft J, Tarjan R (1973) Algorithm 447: efficient algorithms for graph manipulation. CACM 16(6):372–378

Humphries M, Gurney K, Prescott T (2006) The brainstem reticular formation is a small-world, not scale-free, network. In: Biological Sciences, vol 273, p 503511

Kim YH, Seo S, Ha YH, Lim S, Yoon Y (2013) Two applications of clustering techniques to twitter: Community detection and issue extraction. Discrete Dynamics in Nature and Society 2013:1–9

Kleinberg J (2002) Bursty and hierarchical structure in streams. In: KDD, pp 91–101

Kraft T, Wang D, Delawder J, Dou W, Li Y, Ribarsky W (2013) Less after-the-fact: Investigative visual analysis of events from streaming twitter. In: LDAV, pp 95–103

Kwak H, Lee C, Park H, Moon S (2010) What is Twitter, a social network or a news media? In: WWW, pp 591–600

Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. Biometrics pp 159–174

Lappas T, Arai B, Platakis M, Kotsakos D, Gunopulos D (2009) On burstiness-aware search for document sequences. In: KDD, pp 477–486

Lau JH, Collier N, Baldwin T (2012) On-line trend analysis with topic models: #twitter trends detection topic model online. In: COLING, pp 1519–1534

Lee P, Lakshmanan LV, Milios E (2013) Keysee: Supporting keyword search on evolving events in social streams. In: KDD '13, pp 1478–1481

Li C, Sun A, Datta A (2012) Twevent: Segment-based event detection from tweets. In: CIKM, pp 155–164

Marcus A, Bernstein MS, Badar O, Karger DR, Madden S, Miller RC (2011) Twitinfo: aggregating and visualizing microblogs for event exploration. In: CHI '11, pp 227–236

Mathioudakis M, Koudas N (2010) Twittermonitor: trend detection over the twitter stream. In: SIGMOD, pp 1155–1158

McMinn AJ, Moshfeghi Y, Jose JM (2013) Building a large-scale corpus for evaluating event detection on twitter. In: CIKM, pp 409–418

Newman MEJ (2006) Modularity and community structure in networks. Proceedings of the National Academy of Sciences of the USA 103(23):8577–8582

Oxford UP (2009) OUP dictionary team monitors twitterer and tweets. `http://blog.oup.com/2009/06/oxford-twitter/`

Parikh R, Karlapalem K (2013) Et: events from tweets. In: WWW (companion volume), pp 613–620

PearAnalytics (2009) Twitter study. Tech. rep., `http://www.pearanalytics.com/wp-content/uploads/2012/12/Twitter-Study-August-2009.pdf`

Romero D, Chenhao T, Ugander J (2013) On the interplay between social and topical structure. In: ICWSM, pp 516–525

Romero DM, Meeder B, Kleinberg J (2011) Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In: WWW, pp 695–704

Shamma DA, Kennedy L, Churchill EF (2011) Peaks and persistence: modeling the shape of microblog conversations. In: CSCW, pp 355–358

Valkanas G, Gunopulos D (2013) How the live web feels about events. In: CIKM, pp 639–648

Weng J, Lee BS (2011) Event detection in twitter. In: ICWSM, pp 401–408

Yang J, Leskovec J (2011) Patterns of temporal variation in online media. In: WSDM, pp 177–186

Yuheng H, Ajita J, Dorée DS, Fei W (2012) What were the tweets about? topical associations between public events and twitter feeds. In: ICWSM, pp 154–161