Mention-anomaly-based Event Detection and Tracking in Twitter

Conference Paper · August 2014

DOI: 10.1109/ASONAM.2014.6921613

Adrien Guille
Université Lumiere Lyon 2

Cécile Favre
Université Lumiere Lyon 2

Preprint - To appear at ASONAM 2014 (IEEE/ACM International Conference on Advances in Social Network Analysis and Mining)

Mention-anomaly-based Event Detection and Tracking in Twitter

Adrien Guille
ERIC Lab, University of Lyon 2
adrien.guille@univ-lyon2.fr

Cécile Favre
ERIC Lab, University of Lyon 2
cecile.favre@univ-lyon2.fr

***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 method that leverages the creation frequency of dynamic links (*i.e.* mentions) that users insert in tweets to detect important events and estimate the magnitude of their impact over the crowd. The main advantages of *MABED* over prior works are that (i) it relies solely on tweets, meaning no external knowledge is required, and that (ii) it dynamically estimates the period of time during which each event is discussed rather than assuming a predefined fixed duration. 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. Last, we show that *MABED* helps with the interpretation of detected events by providing clear and precise descriptions.**

I. 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 interesting events. These facts raise the following question: *How can we use Twitter for automated 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 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 [1]. In particular, event detection methods focus on detecting "bursty" patterns – which are intuitively assumed to signal events [2] – using various approaches ranging from term-frequency-based approaches [3], [4] to topic-modeling-based approaches [5], [6], including clustering-based approaches [7]–[9]. 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 non-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 *mention-anomaly-based* method, named *MABED* (Mention-Anomaly-Based Event Detection), that can be used in either offline or online settings. *MABED* 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. We claim that we can identify the events that most interest the crowd with a higher accuracy and precision by considering their impact on the mentioning behavior of general users. Our approach also differs from the literature in that it relies solely on statistical measures computed from tweets, whereas existing methods tend to align tweets with external sources of information (*e.g.* Wikipedia, traditional media) in order to reduce noise and improve the accuracy of the detected events. This strategy isn't always appropriate – especially concerning controversial events or politics related events – because it can distort the way events are reported on Twitter. Rather than aligning tweets with external knowledge, our approach intends to reduce noise through a better temporal precision, in that event duration is dynamic whereas most existing methods assume a predefined fixed duration for all events. This finer modeling of bursts helps *MABED* filter out non-related tweets, thus selecting the most relevant words that describe each event, and provides a clearer view of when those real-world events happened.

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. The effectiveness of the method is amplified by the fact that it processes raw tweets, meaning that it requires no time consuming preprocessing (*e.g.* stemming, n-gram identification). To study precision and recall, we ask human annotators to judge whether the detected events are meaningful and significant real-world events. We empirically 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.

Application *MABED* has been used since December 2013 to continuously analyze the French political conversation on Twitter. Based on tweets collected in real-time using the Twitter streaming API, it continuously identifies and keeps track of the most impactful events and helps in understanding how public opinion forms and diffuses on Twitter. The implementation of *MABED* is available for re-use and future research[1]. It includes several user interfaces for exploring events, such as timelines and charts plotting the impact of events through time. *MABED* is also included in *SONDY* [10], a toolkit for mining social data that incorporates several state-of-the-art algorithms.

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 III. Then an experimental study showing the method's effectiveness and efficiency is presented. Finally, we conclude and discuss future work in Section V.

II. 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 studies time gaps between messages in order to detect bursts of email messages [2]. Assuming that all messages are about the same topic, he proposes to model bursts with hidden Markov chains. In [11], authors propose *OLDA* (On-line Latent Dirichlet Allocation), a dynamic topic model based on *LDA* [12]. It builds evolutionary matrices translating the evolution of topics detected in a textual stream through time, from which events can be identified. Authors in [13] 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.

A. Event Detection from Tweets

Term-weighting-based approaches The *Peakiness Score* [3] 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, authors in [4] 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 In [5], authors 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. Hu et al. note that topic modeling methods behave badly when applied to short documents such as tweets [6]. 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. What is more, the study presented in [14] reveals that dynamic topic models don't effectively handle social streams in which several events are reported in parallel.

Clustering-based approaches *EDCoW* [7] breaks down the frequency of individual 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 [8] and [9], 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* [8] 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*, 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.

III. Proposed Method

In this section, we first formulate the problem we intend to solve. Then we give an overview of the solution we propose,

[1] http://mediamining.univ-lyon2.fr/people/guille/mabed.php

TABLE I. TABLE OF NOTATIONS.

Notation	Definition
N	Total number of tweets in the corpus
N^i	Number of tweets in the i^{th} time-slice
N^i_t	Number of tweets in the i^{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^{th} time-slice

MABED, before describing it formally.

A. 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 I 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.

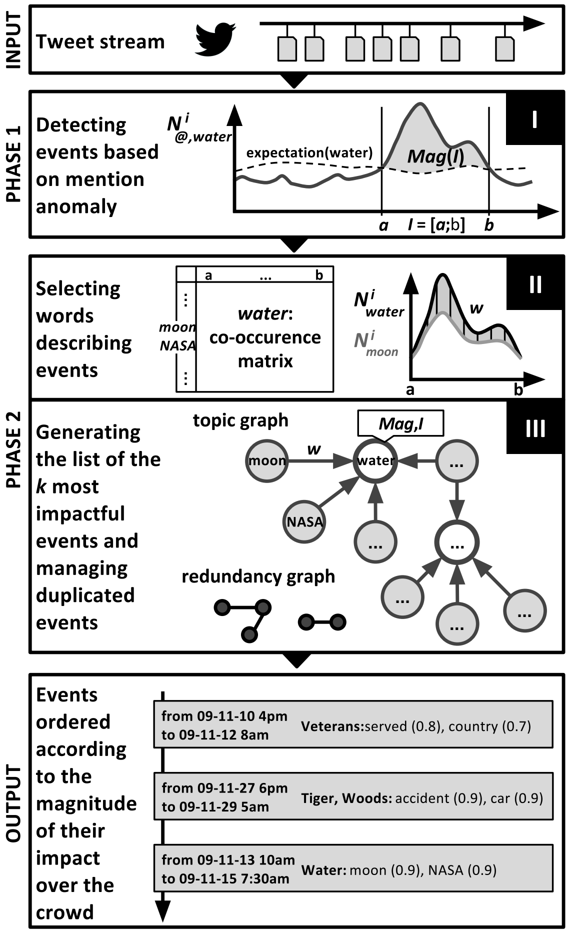
Definition 1 (Bursty Topic): Given a time interval I, a topic T is considered bursty if it has attracted an uncommonly high level of attention during this interval in comparison to the rest of the period of observation. The topic T is defined by a main term t and a set S of weighted words describing 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.

B. 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.

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.



INPUT
Tweet stream
PHASE 1
Detecting events based on mention anomaly
$N^i_{@,water}$
expectation(water)
Mag(I)
a
I = [a;b]
b
I
PHASE 2
Selecting words describing events
a
...
b
moon
NASA
water: co-occurence matrix
N^i_{water}
N^i_{moon}
w
a
b
II
Generating the list of the k most impactful events and managing duplicated events
topic graph
Mag,I
moon
w
water
NASA
...
redundancy graph
III
OUTPUT
Events ordered according to the magnitude of their impact over the crowd
from 09-11-10 4pm to 09-11-12 8am
Veterans: served (0.8), country (0.7)
from 09-11-27 6pm to 09-11-29 5am
Tiger, Woods: accident (0.9), car (0.9)
from 09-11-13 10am to 09-11-15 7:30am
Water: moon (0.9), NASA (0.9)

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

C. 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 of all the words 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^{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 [13]. 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 [8], 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^{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^{th} time-slice this way:

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

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 $[a; b]$.

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

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 that maximizes the magnitude of impact by solving a "Maximum Contiguous Subsequence Sum" (*MCSS*) type of problem. The *MCSS* problem is well known and finds application in many fields [15], [16]. In other words, for a given word t we want to identify the interval $I = [a; b]$, such that:

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

This formulation permits the anomaly to be negative at some points in the interval, only if it permits extending the interval while increasing the total magnitude, which is a desirable property. More specifically, 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 in [17]. 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, $\mathit{Mag}(t, I)$.

D. Selection of Words Describing Events

Starting from the observation that clustering-based methods can in some cases lead to noisy event descriptions, 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 manually 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 [7].

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 $w_{t'_q}$ for each candidate word t'_q. We propose to estimate this weight from the time-series for N^i_t and $N^i_{t'_q}$ with the correlation coefficient proposed in [18]. 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 $I = [a; b]$:

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

$$\text{where } A_{t, t'_q} = (N^i_t - N^{i-1}_t)(N^i_{t'_q} - N^{i-1}_{t'_q}),$$

$$A^2_t = \frac{\sum_{i=a+1}^{b} (N^i_t - N^{i-1}_t)^2}{b - a - 1}, \text{ and}$$

$$A^2_{t'_q} = \frac{\sum_{i=a+1}^{b} (N^i_{t'_q} - N^{i-1}_{t'_q})^2}{b - a - 1}.$$

This practically corresponds to the first order auto-correlation of the time-series for N^i_t and $N^i_{t'_q}$. The proof that ρ_O satisfies $|\rho_O| \leqslant 1$ using the Cauchy-Schwartz inequality appears in [18]. Eventually, we define the weight of the term t'_q as an affine function of ρ_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 $\forall t'_q \in S$, $w_q > \theta$. The parameters p and θ allow the users of *MABED* to adjust the level of information and detail they require.

E. 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 topic 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 simple undirected graph that is used to represent the relations between the eventual duplicated events, represented by their main words. See Figure 1 for an illustration of these structures.

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 topic graph or not. If it isn't the case, the event is introduced into 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 together hereafter.

Detecting duplicated events The event e_1 is considered to be a duplicate of the event e_0 already stored in the topic 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 σ, $\sigma \in]0;1]$. 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.

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 using the algorithm described in [19]. In each connected component, there is exactly one node that corresponds to an event stored in the topic graph. The definition of this event is updated according to the extra information brought by duplicated events. 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.

TABLE II. 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

IV. Evaluation

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.

A. 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 [1]. This corpus contains a lot of noise and chatter. According to the study presented in [20], 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 GMT. Table II gives further details about each corpus.

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 [9]. *TS* is a normalized frequency metric for identifying n-grams that are related to events [4]. 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

TABLE III. PERFORMANCE OF THE FIVE METHODS ON THE TWO CORPORA.

	Corpus: $\mathcal{C}_{en}$			
Method	Precision	F-measure	DERate	Running-time
MABED	0.775	0.682	0.167	96s
α-*MABED*	0.625	0.571	0.160	126s
ET	0.575	0.575	0	3480s
TS2	0.600	0.514	0.250	80s
TS3	0.375	0.281	0.4	82s

	Corpus: $\mathcal{C}_{fr}$			
Method	Precision	F-measure	DERate	Running-time
MABED	0.825	0.825	0	88s
α-*MABED*	0.725	0.712	0.025	113s
ET	0.700	0.674	0.071	4620s
TS2	0.725	0.671	0.138	69s
TS3	0.700	0.616	0.214	74s

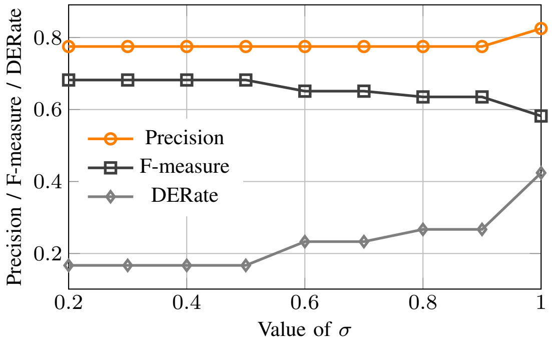


Fig. 2. Precision, F-measure and DERate of *MABED* on $\mathcal{C}_{en}$ for different values of σ.

each time-slice large enough. The maximum number of words describing each event, p, and the weight threshold for selecting relevant words, θ, 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 presented in [21], 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*: σ. In the following, we report results for $\sigma = 0.5$ (we discuss the impact of σ in Section IV-B).

For *ET* and *TS*, because they assume a fixed duration for all events – which corresponds to the length of a 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 asked two human annotators to judge whether the detected events are meaningful and *significant*, by assigning 0 (*i.e.* not significant) and 1 (*i.e.* significant) ratings to each event. The annotators are French graduate students who aren't involved in this project. An event is considered significant if it could be covered in traditional media. Overall, a detected event is significant if it has been rated 1 by both annotators. Considering that both corpora cover a 1-month time period and 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 fraction of detected events that both annotators have rated 1, and recall as the fraction of distinct significant events among all the detected events [8]. We also measure the DERate [8], which denotes the percentage of events that are duplicates among all significant events detected.

B. 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 is $\kappa \simeq 0.76$, showing a strong agreement. Table III reports the precision, the F-measure defined as the harmonic mean of precision and recall, the DERate and the running-time (averaged over three runs) of each method for both corpora.

We notice that *MABED* achieves better performance than α-*MABED* on the two corpora, with an average relative gain of 17.2% in the F-measure. This empirically verifies our main assumption, *i.e.* considering the mentioning behavior of users leads to more accurate detection of significant events in Twitter. Overall, *MABED* yields better performance than all baselines in term of precision and F-measure. *ET* achieves a better DERate on $\mathcal{C}_{en}$, however, this is tempered by the fact that it achieves lower precision and recall then *MABED* on this corpus. Interestingly, *MABED* outperforms all baselines in the F-measure with a bigger margin on $\mathcal{C}_{en}$, which contains a lot more noise than $\mathcal{C}_{fr}$. This suggests that considering the mentioning behavior of users also leads to more robust detection of events from noisy content. Further analysis of the events detected by α-*MABED*, *TS2* and *TS3* reveals that irrelevant events are mainly related to 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 tokens together [22]. In terms of efficiency, we notice that *MABED* and *TS* have running-times of the same order, whereas *ET* is orders of magnitude slower. 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 III don't include the extra time required for indexing data and preparing vocabularies.

Impact of σ on *MABED* While the list of events is constructed by *MABED*, the overlap threshold σ controls the sensitivity to duplicated events. Figure 2 plots the precision, F-measure and DERate of *MABED* on $\mathcal{C}_{en}$ for different values of σ. We observe that the value of σ mainly impacts the DERate. More specifically, the DERate increases along the increase of σ 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 σ 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 $\mathcal{C}_{en}$.

C. Qualitative Evaluation

Next, we qualitatively analyze the results of *MABED* and show how they provide relevant information about the detected

TABLE IV. Top 20 events with highest magnitude of impact over the crowd, detected by *MABED* in $\mathcal{C}_{en}$. Main words are in bold.

#	Time interval (GMT)	Topic
1	from 25 09:30 to 28 06:30	**thanksgiving, turkey**: hope (0.72), happy (0.71) *Twitter users celebrated Thanksgiving*
2	from 25 09:30 to 27 09:00	**thankful**: happy (0.77), thanksgiving (0.71) *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) *Twitter users celebrated the Veterans Day that honors people who have served in the U.S. Armed Forces*
4	from 26 13:00 to 28 10:30	**black**: friday (0.95), amazon (0.75) *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) *The House of Representatives passed the health care reform bill on November 7, 2009*
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) *The Fort Hood shooting was a mass murder that took place in a U.S. military post on November 5, 2009*
7	from 19 04:30 to 21 02:30	**chrome**: os (0.95), google (0.87), desktop (0.71) *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) *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) *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) *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) *Starting from November 10, LinkedIn offered users the possibility to sync their status updates 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) *The Yankees baseball team defeated the Phillies to win their 27th World Series on November 4, 2009*
13	from 15 09:00 to 17 23:30	**obama**: chinese (0.75), barack (0.72), twitter (0.72), china (0.70) *During a visit to China 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) *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) *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) *Related to event #5*
17	from 11 03:30 to 13 08:30	**facebook**: app (0.74), twitter (0.73) *No clear corresponding event*
18	from 18 14:00 to 21 03:00	**whats**: happening (0.76), twitter (0.73) *Twitter started asking "What's happening?" instead of "What are you doing?" from November 18, 2009*
19	from 20 10:00 to 22 00:00	**cern**: lhc (0.86), beam (0.79) *On November 20, proton beams were successfully circulated in the ring of the LHC (CERN) for the 2nd time*
20	from 26 08:00 to 26 15:30	**icom**: lisbon (0.99), roundtable (0.98), national (0.88) *The I-COM roundtable about market issues in Portugal took place on November 26, 2009*

events. Table IV lists the top 20 events[2] 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.

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 (# 6), Chrome (# 7), Tiger Woods (# 8), Obama (# 13). This favors a quicker understanding of events by putting into light the key places/products/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 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 6th event corresponds to Twitter users reporting the Fort Hood shooting that, according to Wikipedia[3], happened on November 5, 2009 between 13:34 and 13:44pm CST (*i.e.* 19:34 and 19:44 GMT). The burst of activity engendered by this event is first detected by *MABED* in the time-slice covering the 19:30-20:00 GMT period. *MABED* gives the following description:

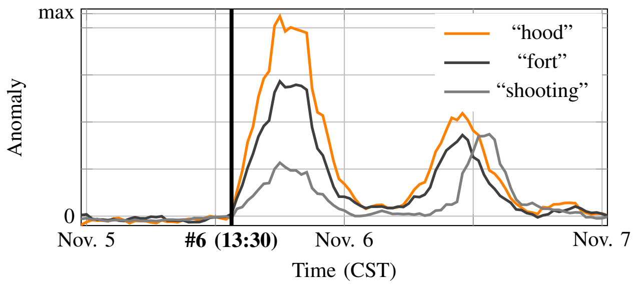


Fig. 3. Measured anomaly for the words "hood", "fort" and "shooting" between Nov. 5 and Nov. 7 midnight (CST).

(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

[2] Due to page limitation, only the top 20 events are listed. More results can be consulted at http://mediamining.univ-lyon2.fr/people/guille/mabed.php

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

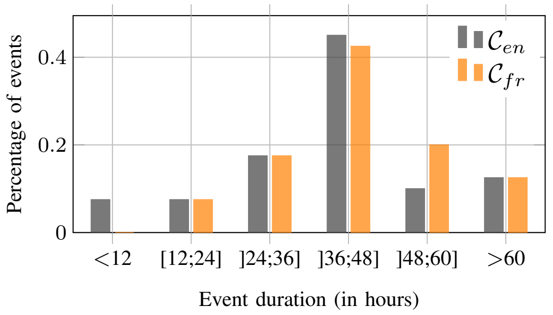


Fig. 4. Distribution of the duration of the events detected with *MABED*.

7:30pm GMT, (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 #1, 5, 6, 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 #6 to illustrate that. Figure 3 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, 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 4, which shows the distributions of event duration for both corpus. It reveals that they follow a normally distributed pattern and that some events are discussed during less than 12 hours whereas some are discussed for more than 60 hours. We notice that the politics related events detected in $\mathcal{C}_{fr}$ tend to be discussed for a longer time than the events detected in $\mathcal{C}_{en}$. This is consistent with the empirical study presented in [23], which states that controversial and more particularly political topics are more persistent than the other topics on Twitter.

V. Conclusion

We described *MABED*, an efficient novel mention-anomaly-based method for event detection and tracking 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 demonstrated the relevance of our approach. Quantitatively speaking, *MABED* yielded better performance in all our tests than α-*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 description 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 detection of duplicated events. 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 geolocation).

Acknowledgments The authors would like to thank Matthew Saltz 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

[1] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *WSDM*, 2011, pp. 177–186.

[2] J. Kleinberg, "Bursty and hierarchical structure in streams," in *KDD*, 2002, pp. 91–101.

[3] D. A. Shamma, L. Kennedy, and E. F. Churchill, "Peaks and persistence: modeling the shape of microblog conversations," in *CSCW*, 2011, pp. 355–358.

[4] J. Benhardus and J. Kalita, "Streaming trend detection in twitter," *IJWBC*, vol. 9, no. 1, pp. 122–139, 2013.

[5] J. H. Lau, N. Collier, and T. Baldwin, "On-line trend analysis with topic models: #twitter trends detection topic model online," in *COLING*, 2012, pp. 1519–1534.

[6] H. Yuheng, J. Ajita, D. S. Dorée, and W. Fei, "What were the tweets about? topical associations between public events and twitter feeds," in *ICWSM*, 2012, pp. 154–161.

[7] J. Weng and B.-S. Lee, "Event detection in twitter," in *ICWSM*, 2011, pp. 401–408.

[8] C. Li, A. Sun, and A. Datta, "Twevent: Segment-based event detection from tweets," in *CIKM*, 2012, pp. 155–164.

[9] R. Parikh and K. Karlapalem, "Et: events from tweets," in *companion WWW*, 2013, pp. 613–620.

[10] A. Guille, C. Favre, H. Hacid, and D. Zighed, "Sondy: An open source platform for social dynamics mining and analysis," in *SIGMOD*, 2013, pp. 1005–1008.

[11] L. AlSumait, D. Barbará, and C. Domeniconi, "On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking," in *ICDM '08*, 2008, pp. 3–12.

[12] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *JMLR*, vol. 3, pp. 993–1022, 2003.

[13] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu, "Parameter free bursty events detection in text streams," in *VLDB*, 2005, pp. 181–192.

[14] L. M. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Goker, Y. Kompatsiaris, and A. Jaimes, "Sensing trending topics in twitter," *IEEE TM*, vol. 15, no. 6, pp. 1–15, 2013.

[15] T.-H. Fan, S. Lee, H.-I. Lu, T.-S. Tsou, T.-C. Wang, and A. Yao, "An optimal algorithm for maximum-sum segment and its application in bioinformatics," in *CIAA*, 2003, pp. 251–257.

[16] T. Lappas, B. Arai, M. Platakis, D. Kotsakos, and D. Gunopulos, "On burstiness-aware search for document sequences," in *KDD*, 2009, pp. 477–486.

[17] J. Bentley, "Programming pearls: algorithm design techniques," *CACM*, vol. 27, no. 9, pp. 865–873, 1984.

[18] O. Erdem, E. Ceyhan, and Y. Varli, "A new correlation coefficient for bivariate time-series data," in *MAF*, 2012, pp. 58–73.

[19] J. Hopcroft and R. Tarjan, "Algorithm 447: efficient algorithms for graph manipulation," *CACM*, vol. 16, no. 6, pp. 372–378, 1973.

[20] PearAnalytics, "Twitter study," www.pearanalytics.com/wp-content/uploads/2012/12/Twitter-Study-August-2009.pdf, Tech. Rep., 2009.

[21] O. U. Press, "OUP dictionary team monitors twitterers tweets. http://blog.oup.com/2009/06/oxford-twitter/," 2009.

[22] G. Valkanas and D. Gunopulos, "How the live web feels about events," in *CIKM*, 2013, pp. 639–648.

[23] D. M. Romero, B. Meeder, and J. Kleinberg, "Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter," in *WWW*, 2011, pp. 695–704.