Data Min Knowl Disc (2016) 30:372–402
DOI 10.1007/s10618-015-0412-3

Efficient temporal mining of micro-blog texts and its application to event discovery

Giovanni Stilo[1] · Paola Velardi[1]

Received: 26 May 2014 / Accepted: 4 March 2015 / Published online: 21 May 2015

Abstract In this paper we present a novel method for clustering words in micro-blogs, based on the similarity of the related temporal series. Our technique, named SAX*, uses the Symbolic Aggregate ApproXimation algorithm to discretize the temporal series of terms into a small set of levels, leading to a string for each. We then define a subset of "interesting" strings, i.e. those representing patterns of collective attention. Sliding temporal windows are used to detect co-occurring clusters of tokens with the same or similar string. To assess the performance of the method we first tune the model parameters on a 2-month 1 % Twitter stream, during which a number of world-wide events of differing type and duration (sports, politics, disasters, health, and celebrities) occurred. Then, we evaluate the quality of all discovered events in a 1-year stream, "googling" with the most frequent cluster n-grams and manually assessing how many clusters correspond to published news in the same temporal slot. Finally, we perform a complexity evaluation and we compare SAX* with three alternative methods for event discovery. Our evaluation shows that SAX* is at least one order of magnitude less complex than other temporal and non-temporal approaches to micro-blog clustering.

Keywords Event detection · Temporal mining · Symbolic Aggregate ApproXimation · Microblog analysis

Responsible editor: Eamonn Keogh.

Electronic supplementary material The online version of this article (doi:10.1007/s10618-015-0412-3) contains supplementary material, which is available to authorized users.

✉ Paola Velardi
velardi@di.uniroma1.it

Giovanni Stilo
stilo@di.uniroma1.it

[1] Department of Computer Science, Sapienza University of Roma, Via Salaria 113, Rome, Italy

1 Introduction

Tracking and classifying events from social media has recently received attention from many researchers (Yan et al. 2013; Weng et al. 2011; Lee and Sumiya 2010; Chae et al. 2013; Pohl et al. 2012; Rui et al. 2012; Dou et al. 2012; Li et al. 2012), even though a recent study (Petrovic et al. 2013) demonstrates that Twitter reports the same events as newswire providers, plus a long tail of minor and mostly irrelevant episodes. Neither stream truly anticipates the other, except for, in some cases and for a few hours, tragic events like earthquakes or terrorist attacks. Despite these findings there are still strong reasons for maintaining a keen interest in micro-blog event analysis: for example, for analyzing the evolution of social phenomena over time (Dou et al. 2012), obtaining information on people's opinion about an event (Maynard and Funk 2012), or improving situational awareness and the impact of public policies (e.g in public health (Dredze 2012)).

Event tracking is often modeled as a problem of topic detection, typically using topic models like Latent Dirichlet Allocation LDA (Blei et al. 2003). Messages are modeled as mixtures of topics, where a topic is a probability distribution over words. To cope with the limited context of Twitter messages various strategies have been adopted, such as aggregating tweets published by the same user (Weng et al. 2010) aggregating tweets using the same word (Hong and Davison 2010), or, more recently, using a "bi-term" (unordered word pairs) model (Yan et al. 2013).

Difficulty in capturing "good" topics in short messages is, however, only one of the shortcomings of topic models:

- First, in LDA, the number of topics is a user-selected parameter, and since this number depends on the document collection many topics are needed in order to obtain fine-grained results. For example, in Chae et al. (2013) an experiment is conducted showing that good results are only obtained with 100 topics and 1000 iterations, a number that quickly becomes prohibitive on a realistically large Twitter collection. In fact, experiments in Yan et al. (2013) are conducted on a very limited sample extracted from the Twitter 2011[1] collection. The authors remark that: "*for very large dataset and a large topic number K, LDA is susceptible to memory problems*".
- Secondly, although an event can be modeled with a topic, e.g. a cluster of terms, not all topics are events. Only a few papers consider this crucial issue. In Dou et al. (2012) an event in the context of social media is defined as "*an occurrence causing change in the volume of text data that discuss the associated topic at a specific time*". In other words, there are *temporal and frequency components* in event detection that need to be analyzed in order to correctly discriminate events from non-events.

In this paper we present an efficient algorithm, named SAX*, for event discovery and tracking from large micro-blog streams. SAX* is based on transforming word temporal series into a string of symbols, using Symbolic Aggregate ApproXimation (SAX). We combine three well-studied methods: SAX, regular expression learning,

[1] http://trec.nist.gov/data/tweets/ sampled from Jan 23rd to Feb 8th, 2011.

and hierarchical clustering, thereby creating an intuitive framework for event detection in microblogs. First, we discretize the temporal series of terms to a small set of levels, which leads to a string for each term; then, a regular expression (regex) is learned from a set of known events in order to differentiate event-like terms from non-event terms; finally, a clustering algorithm is applied on top of the event-like strings to generate clusters which are identified as events. Our method is shown to offer notable advantages over previous methods in terms of computational complexity—a crucial requirement when dealing with large micro-blog streams—and also quality. Complexity is computed in a formal way, as a function of all system parameters, including dimension of the vocabulary and temporal granularity. Quality evaluation is more empirical, as for all previous studies, and is limited by the availability of a lengthy, but poorly dense (1 %) Twitter stream. However, we systematically analyze the performance of our method under different settings of the model parameters, so as to gain an insight into their influence on the type and quality of event that can reliably be detected. We also propose a more objective way for evaluating detected events, which involves searching for related Google news in the same temporal slot.

The paper is organized as follows: in Sect. 2 we summarize the state of the art on temporal mining. Section 3 describes our algorithm in detail. Section 4 is dedicated to a manifold performance evaluation: in Sect. 4.1 we evaluate the complexity of SAX* and we compare it with three other popular temporal and topic-based models for event detection. Then (Sect. 4.2), we tune the model parameters (number of levels, granularity of discretization, etc.) on 8 different world-wide events occurring between July 1st, to August 30th, 2014. Finally, in Sect. 4.3 we evaluate the quality of all discovered events in a 1-year stream from June 2012 to May 2013. Section 5 is dedicated to concluding remarks and future work.

2 Related work

Temporal text mining (TTM) has been defined as the task of "discovering temporal patterns in text information collected over time" (Mei and Zhai 2005). Only recently, however, have temporal mining techniques been proposed for analyzing patterns of collective attention in micro-blog texts (Lehmann et al. 2012; Xie et al. 2013; Weng et al. 2011; Yang and Leskovec 2011). Lehmann et al. (2012) study the evolution of term popularity over time and identify different categories of temporal trends, for example, those in which the activity is concentrated before, after or symmetrically around an event. They then provide a semantic categorization of different patterns of collective attention, using the WordNet[2] lexico-semantic database. Similarly, Yang and Leskovec (2011) detect six common temporal shapes of Twitter hashtags using K-Spectral Centroid clustering. Both papers found that certain "media" or social events have a relatively slow but lengthy acceleration phase and a more rapid decay, while other event types have a long-lasting plateau and/or slow deceleration.

Temporal event mining in micro-blogs is considered in a limited number of papers. In Weng et al. (2011) a temporal analysis technique, named wavelet analysis or

[2] http://wordnet.princeton.edu.

EDCoW, is used to discover events in Twitter streams. As a first step, signals are built for individual words by applying wavelet analysis on the frequency-based raw signals of the words. Autocorrelation is applied to measure the bursty energy of each word. Then, cross-correlation between each pair of bursty words is measured. Finally, a cross-correlation table is used to build a graph, and graph-partitioning techniques are applied to discover relevant events. In Xie et al. (2013) a technique named TopicSketch is proposed for performing real-time detection of events in Twitter. As for EDCoW, events are characterized as "bursty topics", i.e. a set of words showing a sudden surge of popularity followed by a decay. TopicSketch computes in real-time the acceleration (the second order derivative) of three quantities: a) the total Twitter stream; b) each word in the stream; c) each pair of words in the stream. Given these (known) quantities, the distributions of words over a set of bursty topics is estimated by modeling the mixture of multiple inhomogeneous processes of topics as a Poisson process, and then solving an optimization problem. Hashing techniques and process parallelization are used to keep the problem tractable in terms of memory cost and computational complexity. In fact, one of the main problems with TTM when applied to large and lengthy data streams is its computational cost.

Both the foregoing methods assume that events are invariably of a "bursty" character, whereas, as shown by Lehmann et al. (2012) and Yang and Leskovec (2011), a variety of different event-patterns exist. Another often ignored issue is the impact of temporal discretization on computational tractability. For example, the authors of TopicSketch demonstrate empirically that their method is computationally tractable under specific conditions, e.g. discretizing the time line in slots Δ of 15 min to compute acceleration, analyzing tweets at a local scale (in Singapore) to keep the number of potential topics low,[3] and distributing the computation on multiple cores. In the absence of a formal complexity analysis, it is hard to estimate the actual processing requirements of their method under different conditions. In fact, the authors observe that "*the more bursty the event is, the better TopicSketch performs*": however the "burstyness" of events depends upon the size of Δ, a parameter that does indeed impact on performance: for example, a larger Δ considerably increases the number of "active" words to be traced by the algorithm. Probably, larger values of Δ (e.g. 1 h or 1 day) would allow TopicSketch to detect phenomena with slower acceleration, like those analyzed in Lehmann et al. (2012), but this would increase complexity.

As discussed in the Introduction, a much larger number of papers have been concerned with the task of event detection, tracing or discovery, without using temporal mining. State-of-the-art approaches to event detection are based on n-grams clustering techniques and/or latent topic detection, such as LDA (Blei et al. 2003) with its many variants (Yan et al. 2013; Ifrim et al. 2014; Lee and Sumiya 2010; Chae et al. 2013; Pohl et al. 2012; Rui et al. 2012; Dou et al. 2012; Li et al. 2012; Wang et al. 2013; Huang et al. 2012; Popescu et al. 2011). Among these systems, one of the most recent and widely cited is BiTerm LDA (Yan et al. 2013), in which, first, consecutive word pairs, named bi-terms, are extracted, following which a topic model is then applied

[3] In any case a limit is fixed a priori for the number of topics.

to extract relevant topics. We note that these methods can only work in an “off-line” manner, since they ignore the temporal aspect of the phenomena they aim to track.

Finally, some papers propose a mixed approach, in which both lexical and temporal features are considered. In Li et al. (2012) a system named Twevent is proposed. Like for BiTerm LDA, tweets are first segmented to extract relevant word pairs, and a timestamp is associated to each pair. Then, bursty pairs are segmented and clustered. In Dao et al. (2012) and Hong et al. (2011) a topic model is used to generate topics and, in a subsequent step, the authors identify bursty topics. A very recent paper by Cheng and Wicks (2014), proposes identifying space-time location of Twitter clusters detected with LDA. These papers suffer from both the problems mentioned hitherto: computational complexity of feature extraction from tweets, and simplified temporal patterns.

A final problem, shared by all papers dealing with event detection, is evaluation. We note that the unavailability of common, independently annotated datasets is a serious shortcoming of all current approaches to Twitter mining, since it prevents proper comparison between different methods. On the other hand, virtually all the temporal mining methods surveyed in this paper up to this point evaluate detected events in a purely qualitative manner, labeling them manually, and often without even providing a measure of precision. This will be discussed in the evaluation Sect. 4.

3 “Time makes sense”: temporal similarity as a measure of semantic relatedness

In this Section we describe the SAX* algorithm and its application to term clustering. The underlying idea of SAX* is that words with a similar temporal behavior are related to each other. The nature of this relatedness is either limited to a specific temporal slot, e.g. when terms describe a unique event, or is more systematic and repetitive, for example when terms refer to periodic, culturally related, issues (such as holy days, social rituals, collective deadlines, etc.). In this paper we are more interested in the first case, i.e. event-related clusters. To tune and evaluate our approach we collected 1 % of Twitter traffic, the maximum freely allowed traffic stream, from January 2012 to August 2014, using the standard Twitter API.[4] We used part of this stream for learning the temporal behavior of terms related to events (Sect. 3.2) and parameter tuning (Sect. 4.2), and part for testing. We stemmed and indexed only tweets in English, removing tweets with a hyperlink and removing words of less than three characters. Our dataset, hereafter referred to as the *1 % Twitter stream*, consists of about 3500 million tweets (about 160 million of which are in English).This is considerably bigger than the 250 million tweets of the Twitter 2013 collection, which was, to the best of our knowledge, hitherto the largest available collection used in micro-blog analysis. What is more, the Twitter 2013 collection spanned only 2 months. We considered a longer time span was indispensable in order to track a sufficiently wide variety of events. Another large collection is described in Cha et al. (2010), however this collection was

[4] https://dev.twitter.com/docs/streaming-apis.

obtained (with the permission of Twitter) by crawling only about 80,000 Twitter users and, moreover, is not available.

Our SAX* approach is based on three steps, which will be described in the next three Sections:

1. First, word[5] streams in a temporal window W are reduced to a string of symbols, using SAX (Lin et al. 2007, 2012).
2. Then, a regular expression is learned to identify strings representing patterns of collective attention.
3. Finally, hierarchical clustering with complete linkage is used to group tokens with similar strings occurring in the same window.

3.1 Symbolic aggregate approXimation

SAX (Lin et al. 2007) allows a time series of arbitrary length W to be reduced to a string of arbitrary length N, (typically $N << W$). Given a time series $S(t)$, this is discretized using a well-defined dimensionality reduction method called Piecewise Aggregate Approximation (Keogh et al. 2001). The PAA representation is as follows: given the function $S(t)$ in a window W, this can be discretized into N partitions of equal length Δ (e.g. days, hours..). We denote with $\bar{s}_i$ (i $= 1 \ldots N$, $N = \frac{W}{\Delta}$) the mean value of $S(t)$ falling into each partition i. Then, the PAA representation is symbolized with a discrete string, using an alphabet $\Sigma : \{a, b, ..\}$ of $|\Sigma| = n$ symbols. Since normalized time series have a highly Gaussian distribution, we can determine the breakpoints β_j ($j = 1..n - 1$) that produce n equally sized areas under the Gaussian curve. Once the breakpoints have been established, the PAA representation is turned into a string of symbols in the following way:

$$\hat{s}_i = j, \quad j \in \Sigma, \quad iff\, \beta_{j-1} < \bar{s}_i < \beta_j$$

The SAX representation significantly reduces the dimensionality of data, but also its numerosity for some applications, like event detection, as will be discussed later. Furthermore, Lin et al. (2007) show that a distance measure between two symbolic strings lower bounds the true distance between the original signals, which is a key result justifying the use of string clustering rather than signal clustering.

For our task of event detection we first normalize all tokens through z-score normalization[6] (though other normalization techniques could also be adopted) and we then apply PAA discretization. We note that the system parameters $|\Sigma|$, W and Δ have an influence on the type and granularity of the events to be detected. Intuitively, local events are better detected within smaller slots (e.g. 15 min, like in *TopicSketch* (Xie et al. 2013)), while a coarse discretization Δ and window W should capture

[5] Words are stemmed to reduce sparseness, even though, as discussed in the paper, this might not be strictly necessary with more dense Twitter streams. In what follows we will refer to clustered items interchangeably as words, stems, or tokens.

[6] http://code.google.com/p/jmotif/wiki/ZNormalization.

world-wide events with a more lasting impact on Twitter users. Concerning $|\Sigma|$, we could argue that larger alphabets allow better demarcation between different pattern categories, e.g. disasters as opposed to media events, as discussed e.g. in Lehmann et al. (2012), while smaller alphabets may provide a higher recall since they impose less restrictive conditions on pattern similarity. In the rest of this Section, based on the consideration that our 1 % world-wide stream is not dense enough to obtain evidence of small, local events, we set $|\Sigma| = 2$, $\Delta = 1$ day and $W = 10$ days. However, the effect of different parameter settings on efficiency is systematically analyzed in Sect. 4.2.

Figure 1 shows the SAX string associated with the normalized time series $S'(t)$ for the word *Ryder*. The series refers to a 10-day window W starting on September 25th, 2012, with a 1-day discretization Δ and binary alphabet. The x axis represents the breakpoint β (with $|\Sigma| = 2$ and z-normalization, there is only one breakpoint at $y = 0$) and the dashed line shows the $\bar{s}_i$ values. Using the binary alphabet $\{a, b\}$, the correspondent SAX string for *Ryder* is *aaaababaaa*.

Figures 2, 3, 4 and 5 illustrate the utility of z-normalization, which is an important step of our algorithm: Fig. 2 shows the time series, in the same window as in Fig. 1, for the word stems: *ryder rydercup europ golf* (co-occurring during the Ryder Cup golf competition), while Fig. 3 shows the same series after z-normalization. These examples demonstrate that, even though the time series do not display identical shapes,

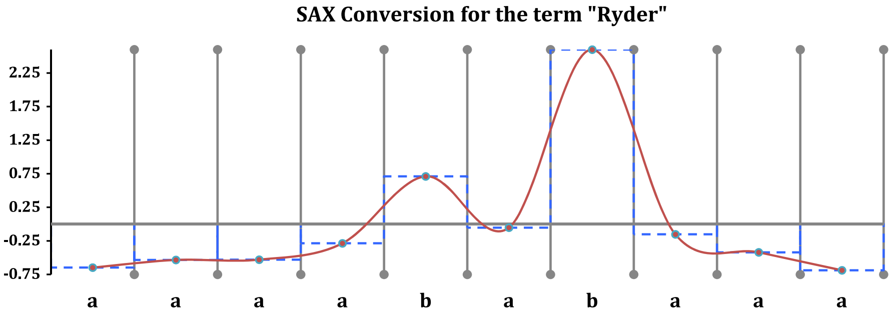


Fig. 1 Binary SAX representation ($\alpha = 2$) of the term "Ryder"

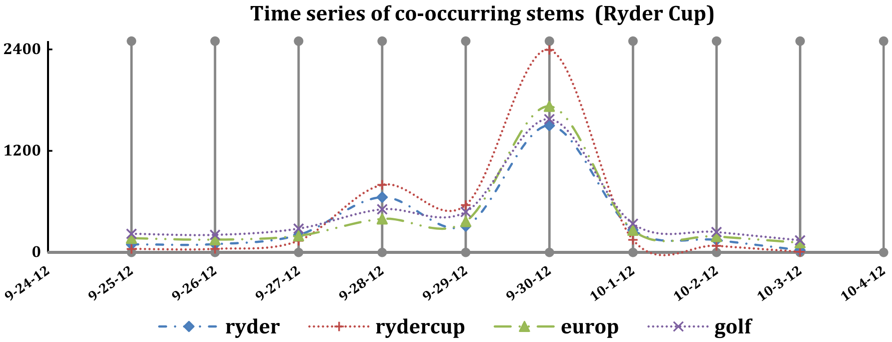


Fig. 2 Non-normalized time series for: ryder, rydercup, europ, golf

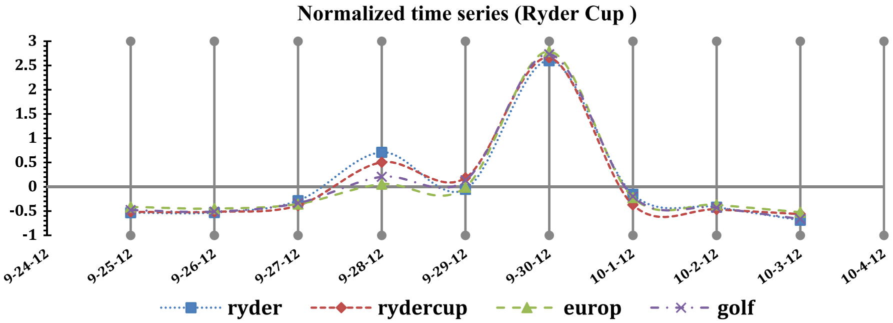


Fig. 3 Normalized time series for: ryder, rydercup, europ, golf

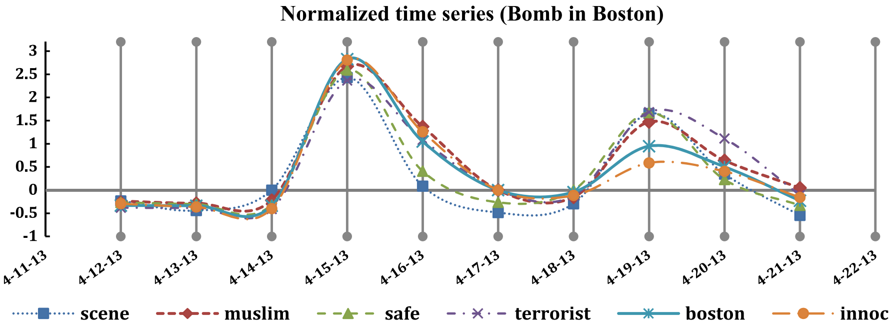


Fig. 4 Normalized time series for: scene, muslim, safe, terrorist, boston, innocent

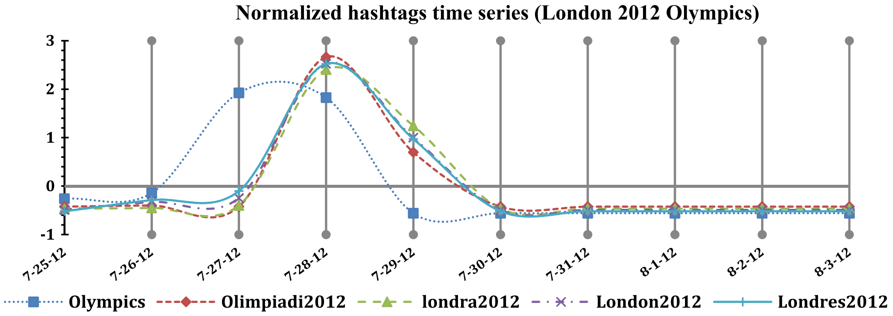


Fig. 5 Normalized time series for hashtags: #Olympics, #Olimpiadi2012, #londra2012, #London2012, #Londres2012

especially before normalization, their corresponding SAX strings are the same or very similar, intuitively suggesting a correlation. Additional examples are in Figs. 4 and 5, showing co-occurring normalized time series during the Boston Marathon event, and a set of multi-lingual co-occurring hashtags during London Olympic Games in 2012.

3.2 Learning patterns of attention

As remarked in the Introduction, an important (and often ignored) task in event detection consists of establishing a suitable temporal characterization of events. Previous work by Lehmann et al. (2012) and Yang and Leskovec (2011) used signal clustering to learn typical shapes of attention in micro-blogs. However, these shapes are shown and discussed in qualitative terms, rather than formally characterized. In Lin et al. (2007) a method is proposed for anomaly detection in signals, based on SAX. The authors use suffix trees and Markov chains to model the "normal" behavior of strings and compare current against "normal" behavior. In our case, however, the "normal" behavior of individual tokens can be very different: for example, it is normal that certain hashtags, like "*#thankgoditsfriday*" spike every week. Since we are interested in detecting clusters of synchronous and "anomalous" signals, a better approach is to learn a generalization of anomalous, rather than normal, behaviors—similarly to Lehmann et al. (2012) and Yang and Leskovec (2011) - although our aim is to characterize the anomalous behavior in a more formal way. With SAX, temporal series associated with words or key phrases are represented as sequences of symbols, therefore the problem of learning anomalous behaviors can be cast in terms of learning regular expressions ("regex").

To learn regex capable of identifying events we trained our system on a subset Wikipedia Events[7] in early 2012, manually selecting a few relevant words within each event description, and tracing their temporal series on a Twitter dataset in a ±5 days window around the main day of the event. In order to select "good" words we also verified on Google Trends[8] that each candidate word had actually had an impact on web users: examples are *Greek debt*, *Queen Elisabeth* (her diamond Jubilee), etc. Each word was then converted into a SAX string alphabet, and kept only if it had at least an $a \rightarrow b$ and $b \rightarrow a$ transition in the window (in the case of binary alphabet). We then used the extracted patterns for all the considered events and we used the RPNI algorithm (Oncina and Garcıa 1992), available in the libalf[9] library, to generate compatible regular expressions. Finally, we learned the following regular expression:

$$(a+b?\,bb?\,a+)?\,(a+\ b?\,bba*)? \tag{1}$$

With an alphabet of 3 symbols, we used a similar procedure to learn the following expression:

$$(a+[\mathrm{bc}]?\,[\mathrm{bc}]\,[\mathrm{bc}]?a+)?\,(a+[bc]?\,[bc]\,[bc]a*)? \tag{2}$$

These regex capture all the series with one ore two peaks and/or plateaus in the analyzed window, such as, for example, the sequences in Figs. 1, 2, 3, 4, and 5. With an alphabet of 3 symbols (or more) we can capture more subtle differences, like slowly versus sharply rising/leading edges, or peaks of different intensity. With larger alphabets, on

[7] http://en.wikipedia.org/wiki/2012.

[8] https://www.google.it/trends/.

[9] http://libalf.informatik.rwth-aachen.de/index.php?page=home.

the other hand, regex become progressively more specific, and so do the similarity constraints on co-occurring signals (see Sect. 4.2 for a more accurate analysis).

Note that the regex (1) and (2) would not be captured by a simple bursty model such as that in Weng et al. (2011). Importantly, these regex turn out to be a generalization of 5 out of 6 shapes of attention learned by the algorithm described in Yang and Leskovec (2011),[10] a fact that can be considered as further evidence of their generality. The shape that we do not capture accurately is the one with three peaks, however the authors do not mention the frequency or probability of this shape. Note that (1) and (2) would in any case capture a multiple peaks stream in a window W, but, depending on the intensity and distance between peaks, these would either be merged, or split into two events.

3.3 Temporal clustering

As suggested by the examples in Figs. 3, 4, 5, our aim is to cluster terms on the basis of the similarity of their time series. The SAX representation enables this similarity to be captured efficiently. To create term clusters we proceed as follows. In our 1 % Twitter stream we consider sliding windows W_i partitioned in slots of length Δ. At each execution of the clustering algorithm the window is shifted by a slot Δ. Within each window W_i, first, we purge terms with a frequency lower that f, then, surviving terms are converted into SAX strings with $|\Sigma|$ symbols. We then consider only those terms matching the learned regex in W_i, thereby greatly reducing the computational and memory requirements of the subsequent clustering phase (this will be discussed in more detail below). Let $L'(W_i)$ be the subset of terms in L that match the regex. Over these terms we apply a bottom-up hierarchical clustering algorithm with *complete linkage* (Jain 2010). The clustering literature is immense, and many other algorithms are available: however, complete linkage avoids the so-called *chaining phenomenon*, which causes one cluster to attract most of the population members. Furthermore, unlike the majority of clustering algorithms, complete linkage is not heavily parametric.[11] In complete linkage the similarity of two clusters is defined as the similarity of the most dissimilar members, which is equivalent to choosing the cluster pairs whose merge has the smallest diameter. The algorithm starts with singleton clusters (e.g. each consisting of one term), and then progressively merges pairs of clusters into larger ones, according to a "smallest diameter" criterion, measured using a selected distance function.[12] We stop hierarchical bottom-up clustering aggregation for a cluster $c_j^{W_i}$ when $SD\,(d(centroid, t_k)) < \delta$, where SD is the standard deviation of the distance between all terms t_k in $c_j^{W_i}$ and the cluster centroid. We further purge clusters smaller than h elements. To summarize, the clustering parameters are f, h and δ. To simplify, we set $h = 3$, therefore the final set of parameters for SAX* is: $W, |\Sigma|, \Delta, f, \delta$.

[10] See Fig. 8 of the mentioned paper, in which 6 shapes of attention of Twitter hashtags are shown.

[11] For example, in many algorithms the number of cluster K is a parameter.

[12] We use the euclidean distance, but other measures, e.g. the edit distance, produce very similar results.

Let $C^{Wi} = \left\{c_1^{Wi}, \ldots c_{ni}^{Wi}\right\}$ be the clustering result in W_i. As an example, consider the 10-day window starting on April 12, 2013 (Fig. 4). In this window we obtain the following three clusters (the corresponding SAX* sequences are shown only for the cluster centroid) with $W = 10$, $|\Sigma| = 2$, $\Delta = 24$hrs, $f = 3000$, $\delta = 0.5$:

c1:[***scene, muslim, safe, terrorist, boston, innoc***] Centroid, SD: (aaabbaabba, 0.4082)

c2:[***purpos, boat, cop, suspect, weed***] Centroid, SD:(aaaaaaabba, 0.4472)

c3:[***bomb,affect,injur,marathon,victim***] Centroid, SD:(aaabbaaaaa, 0.4472)

In this example the three clusters are related to the same event (namely, a bomb during the Boston Marathon), which is the most frequent case given the granularity of one day and the cluster dimension filters. Note also that cluster **c2** represents a late subtopic of the event (the "*b*"s are towards the end of the window), namely, when a terrorist was eventually found hidden in a boat. The example also demonstrates the nice tendency of complete linkage to create balanced clusters.

Finally, we can observe that in this particular case the start and end of the event fall perfectly within the ten days of the temporal window W. However, events are not known *apriori*, therefore this can not be guaranteed in general. This is the reason for using sliding windows: sliding is better than slicing, since if a slice were to cut an event in two we wouldn't be able to detect it. On the other hand, since consecutive windows overlap over 9 days, we may have many windows and many clusters that capture the same event or some of its subtopics. With reference to the previous three clusters, the window W' obtained by sliding W one day to the right, would still generate more or less the same clusters, while sliding six days would capture only **c2** and maybe additional related or unrelated clusters. Therefore, a method is needed for capturing significant events on a day-to-day basis (or, more in general, in Δ slots).

To this end we proceed as follows: For every temporal slot Δ_j consider the set $\boldsymbol{W}$ of all windows such that $\Delta_j \in W_i$. For each $W_i \in \boldsymbol{W}$, select the subset of clusters $C_{\Delta_j}^{W_i}$ in W_i with a peak in Δ_j, e.g., if a binary alphabet is adopted, those whose centroid has a "*b*" in Δ_j. Then, the subset of clusters in Δ_j is: $\mathrm{C}^{\Delta_\mathrm{j}} = \left\{\mathrm{C}_{\Delta_\mathrm{j}}^{\mathrm{W_i}}\right\}$. With reference to the previous window starting on April 12th, on day April 19th the relevant clusters would be **c1** and **c2**.

Note that clustering is performed on sliding windows W_i while Δ-clusters are obtained in a post-processing step: C^{W_i} are the clusters in W_i and $\mathrm{C}^{\Delta_\mathrm{j}}$ is the subset of clusters with a peak in Δ_j.

4 Evaluation

As we remarked in the Introduction the main difficulty with regard to evaluating event detection on micro-blogs is the absence of common annotated datasets. This absence prevents proper comparison between different methods from being undertaken except when a specific algorithm is made available to the community, which is rarely the case.[13]

[13] For example, there are many available implementations of LDA.

Evaluation methods in the literature depend on the type of approach used:

- Methods based on topic/n-gram clustering (see short survey in the Sect. 2) mostly use standard methods for cluster evaluation. For example, the authors of BiTerm LDA (Yan et al. 2013) provide a manifold evaluation: First, they compute a measure of cluster cohesiveness, namely the ratio between intra-cluster and inter-cluster distance, named H-score. They also compute other internal cluster validity indexes, such as Purity, Normalized Mutual Information, and the Adjusted Rand Index. Other topic model approaches use Perplexity as an internal evaluation measure (Hong et al. 2011). However, in Kovacs et al. (2005) it is experimentally shown that none of the commonly proposed internal clustering validity indexes reliably identifies the best clusters, unless these are clearly separated. We also remark that our algorithm pursues co-occurrence in time rather than in texts, therefore the above listed measures cannot be used in a straightforward manner on SAX*.
- Virtually all the approaches based on temporal mining are tested on very limited datasets (e.g. a few days, or relatively few tweets in a specific region or city, e.g. Singapore). As remarked in Petrovic et al. (2010), evaluating on a large scale "*would be prohibitively expensive as it would involve human experts going through 30 million tweets looking for first stories*". All the papers in this domain perform a manual labeling of detected events on a selected subset of different Twitter streams (Xie et al. 2013; Yan et al. 2013; Chae et al. 2013; Pohl et al. 2012; Rui et al. 2012; Dou et al. 2012; Li et al. 2012; Wang et al. 2013; Huang et al. 2012; Popescu et al. 2011; Oncina and Garcıa 1992; Weng et al. 2010; Hong and Davison 2010; Dao et al. 2012). Usually only a limited number of examples of the detected clusters are shown, without (apart from just a few cases) providing explicit measures of precision (Li et al. 2012; Weng et al. 2011). For example, the authors of TopicSketch do not provide a measure of precision, but merely present and analyze a table with 12 examples of clusters generated by TopicSketch and another algorithm named Twevent (Li et al. 2012) using the same dataset of tweets sent by 19,256 unique Singapore users. Since this dataset **is** not annotated with ground truth, it would be of little use to researchers who were unaware of Singapore events and culture.[14] On the other hand, with the exception of the dataset recently presented in McMinn et al. (2013), no datasets are currently available for golden standard event detection in micro-blogs. The McMinn et al. dataset, nevertheless, provides relevance judgment for only about 150,000 tweets, making it more appropriate for algorithms based on text analysis rather than temporal mining.

Table 1 summarizes the evaluation strategy adopted by the most important event detection methods surveyed in Sect. 2.

Another crucial evaluation parameter is computational efficiency. However, only in Yan et al. (2013) is a formal complexity evaluation provided. In our view, complexity needs to be of very serious concern when text processing and clustering algorithms are applied to very large collections of data such as our 1 % Twitter stream, or the 2013 Twitter Collection. Instead, even in the recent TREC 2013 micro-blog track

[14] Some of the events shown in the related papers are world-wide, but several are local events, e.g. "Super Junior's Yesung (@shfly3424) created his Twitter account".

Table 1 Summary of performance strategies adopted in event detection literature

Article	Method	Temporal mining	Method of evaluation	Quantitative evaluation
Mei and Zhai (2005)	Probabilistic mixture model	Yes	Qualitative analysis of extracted events	None
Li et al. (2012) TWEVENT	Tweet segmentation, burstyness probability and clustering	Yes	Qualitative analysis of extracted events	Manual labelling: Precision = 86.1 %
Weng et al. (2011) EDCoW	Wavelet analysis	Yes	Qualitative analysis of extracted events	Manual labelling: Precision = 76.2 %
Rui et al. (2012) TEDAS	Content and social features	No	None	None
Pohl et al. (2012)	Content features, Self Organizing Maps	No	Qualitative analysis of extracted events	None
Xie et al. (2013) TopicSketch	Detection of acceleration in temporal signals	Yes	Qualitative analysis of extracted events	None[a]
Yan et al. (2013) BiTerm LDA	Biwords extraction and latent topic model	no	Internal cluster validity measures	Cluster validity measures H-score = 0.474 ± 0.005

[a] However, the authors compare with Twevent using the same Singapore users dataset

challenge, participants were not evaluated taking formal complexity into account.[15] In this Section we propose a manifold evaluation of our SAX* approach:

- First, we provide a complexity evaluation of SAX*, comparing against our three major competing approaches: BiTerm LDA, EDCoW and TopicSketch. The first was selected for its popularity, since, as already remarked, in literature event detection is often casted in terms of topic detection. The others were selected because they exploit time series similarity, as we do. We show that our method is by far the most advantageous (one or two orders of magnitude) in terms of complexity;
- Second, we use a two-month sub-set of our 1 % Twitter stream that is particularly dense in very diverse world-wide events (sport, politics, disasters, etc), in order to analyze the effect of different choices of the 5 SAX* parameters on the precision and recall of the algorithm.
- Third, we perform a manual evaluation of the events detected along 1 year of our 1 % Twitter stream, using the best parameter setting identified in the previous step. Our evaluation methodology is more "principled" in comparison to previous methodologies, since we objectively verify that there is a match between a detected event and a real event which actually occurred during that same period (provided only

[15] https://github.com/lintool/twitter-tools/wiki/TREC-2013-Track-Guidelines.

that this real event had some “echo” on the web). We believe that our methodology is more appropriate for evaluating competing systems than building annotated corpora (McMinn et al. 2013) since it requires only a straightforward judgment, rather than complex and lengthy manual annotation.

4.1 Complexity evaluation

In this Section we perform a complexity evaluation of SAX*, and we compare it with BiTerm LDA (Yan et al. 2013), EDCoW (Weng et al. 2011) and TopicSketch (Xie et al. 2013). We select BiTerm LDA, even though it does not use temporal mining, for three reasons:

1. Several approaches to event detection use LDA, and BiTerm LDA is one of the most cited and recently published paper in this line;
2. Any other topic model approach with temporal post-processing would exhibit at least the complexity of a standard LDA;
3. Any of the “mixed” models based on bigram extraction and temporal mining (see Sect. 2) would be affected by the same temporal penalty of Biterm LDA in extracting word pair features.

For SAX*, the complexity analysis is based on Lin et al. (2007) and on personal communication with the author; for BiTerm LDA, we use the complexity formula provided by the authors in Yan et al. (2013); for EDCoW and TopicSketch our computation is based on the algorithm description presented in the respective papers (Weng et al. 2011; Xie et al. 2013), which we briefly summarize in what follows. We introduce the following parameters:

$\boldsymbol{D}$	number of tweets in W
$\boldsymbol{t}$	average document (tweet) length
$\boldsymbol{L}$	vocabulary dimension (lexicon) in W
$\boldsymbol{L'}$	vocabulary dimension after pruning (when applicable)
$\boldsymbol{\Theta}$	re-sampling window in EDCoW
$\boldsymbol{W}$	window length
$\boldsymbol{K}$	number of discovered events/topics (this is a manually defined parameter only for BiTerm and TopicSketch)
$\boldsymbol{B}$	number of biterms in BiTerm LDA
$\boldsymbol{H}$	number of hash functions in TopicSketch
$\boldsymbol{I}$	number of iterations of outer loop in TopicSketch
$\boldsymbol{i}$	number of iterations of Newton-Raphson method in TopicSketch

SAX* complexity

The first step requires reading the documents, indexing the terms, and creating a temporal series for every term. Supposing an average length per document of t terms, this step takes Dt.[16] Then, we read the lexicon, pruning all terms below a given frequency, with cost L. Let L' be the pruned lexicon. Finally we remove all terms that

[16] In what follows we omit the “big-o” notation for simplicity: complexity formulas are all to be interpreted as “order of”.

do not match the regex (1), with a cost that is linear in the dimension of the window W: $L'W$. Let L'' be the final dimension of the lexicon. The worst case is when $L' = L''$ though in general $L' \gg L''$. The number of comparisons among symbolic strings during hierarchical clustering with complete linkage depends on the string length, which is $\frac{W}{\Delta} = W$ (since $\Delta = 1$), therefore the worst-case cost is $O((L' - 1)(W^2 L'))$. After the clustering step, K clusters are generated. Finally, we apply cluster pruning—small clusters are removed—with a cost $O(\mathrm{K})$.

To summarize, the cost is :

$$O(Dt + L + L'W + (L' - 1)(W^2 L') + \mathrm{K})$$

Hereafter we will assume that the "*big-O*" notation is implicit, to avoid notational overloading.

EDCoW complexity

A detailed description of the algorithm is found in Weng et al. (2011). As for SAX*, the first step consists of the transformation of terms in documents into temporal series with cost Dt. In the first stage D_1 of the algorithm, every term-related signal s_i is converted into another signal s_i'; the new signal is obtained by applying Shannon Wavelet Entropy to sub-sequences of length Θ of the original signal s_i. In other words a value s_i' is computed every Θ values of s_i. In stage D_2, two contiguous values s_i', s_{i+1}' are aggregated. The cost of the first stage operation is then: $L\left(\frac{W}{\Theta}\left(\Theta^3 + \Theta\right)\right)$. The second stage filters signals s_i' (of length $\frac{W}{\Theta}$) using the autocorrelation function; this part has a cost $L\left(\frac{W}{\Theta}\log\left(\frac{W}{\Theta}\right)\right)$ and produces a sub-lexicon L' .Next, EDCoW builds the cross-correlation matrix for all pairs of remaining terms. The cost needed to build the cross-correlation matrix is $\left(L'\right)^2 \frac{W}{\Theta}^2$.In the subsequent phase EDCoW detects events though modularity-based graph partitioning that is efficiently computed using *power iteration* at cost L'^2. For each event $e \in \mathrm{E}$ ($|\mathrm{E}| = \mathrm{K}$) the final cost is bounded by KL'^2. The final step consists of selecting the events on the basis of their related sub-graph and can be included in the previous phase without additional cost. The total cost of the algorithm is then summarized by the following formula:

$$Dt + L\left(\frac{W}{\Theta}\left(\Theta^3 + \Theta\right)\right) + L\left(\frac{W}{\Theta}log\left(\frac{W}{\Theta}\right)\right) + \left(L'\right)^2 \frac{W}{\Theta}^2 + KL'^2$$

BiTerm complexity

The BiTerm algorithm is based on the LDA algorithm, described in Blei et al. (2003). The only additional step is the initial computation of word pairs (co-occurring terms in tweets), or bi-terms, that are generated at cost $\frac{Dt(t-1)}{2}$. Bi-terms are an essential step of the algorithm proposed in (Yan et al. 2013) since they are shown to considerably improve the quality of detected topics in short texts, such as tweets. As stated in Yan et al. (2013), the cost of generating K clusters with LDA is K(Dt(t − 1)/2). Therefore, the total cost of BiTerm LDA is:

$$\frac{Dt\,(t-1)}{2} + \frac{\mathrm{K(Dt(t-1)}}{2}$$

TopicSketch complexity

In Xie et al. (2013) the authors present a detailed description of the algorithm, though they do not provide a complete complexity analysis. As for the other algorithms, the first step consists of reading the stream and collecting terms statistics with cost Dt. Then a dimension reduction is applied with cost $\mathrm{H}(1 + L/L')$, where H are hash functions mapping words to bucket $[1...L']$[17] uniformly and independently. The cost of the subsequent phase is summarized by the computational cost of maintaining all the $\mathrm{H}t^2$ accelerations (this cost is provided by the authors). The last step is a topic inference algorithm, modeled as an optimization problem. The gradient-based method[18] to optimize the objective function f is based on the Newton-Raphson approach, whose complexity depends on the multiplication function.[19] Using a very conservative value of 32 bit precision the cost is at least: $I \cdot H \cdot K \cdot i \cdot L' \cdot \log(32)$. Though some minor costs are ignored for the sake of simplicity, the final complexity is order of:

$$Dt + \mathrm{H}\left(1 + L/L'\right) + (\mathrm{H}t^2) + (I \cdot H \cdot K \cdot i \cdot L' \cdot \log(32))$$

Complexity estimates

Given the above formulas, we can now provide quantitative complexity estimates. We set the parameters as follows:

- the length t of documents is set to 9.4 words;[20]
- the size of D grows from 100 to 10 million tweets, which is about the actual average size (9,163,437) of English tweets in a 10-day window in a 1 % Twitter stream;
- the vocabulary L grows according to a Zipfian law with parameter $\alpha = 1.127$ estimated on our Twitter stream. L' grows according to the same law (starting from L), with an estimated parameter $\alpha = 0.41456$.
- $\Theta = 3$ as reported in Weng et al. (2011), the window W is 10 days, and Δ =1 day. Note that, in TopicSketch, W indirectly impacts on performance since according to the authors it limits the dimension L of the words to be traced to a manageable value. The impact of W and Δ is accounted for by the cost of maintaining the accelerations, $\mathrm{H}t^2$.
- finally, the number of clusters is set to 50, in accordance with the initial K value chosen in Yan et al. (2013) for the BiTerm LDA algorithm.
- in accordance with Xie et al. (2013) we set H to 6, I to 50 and i to 25.

Table 2 shows that SAX* is one order of magnitude less complex than ECDoW and TopicSketch, and two orders less than BiTerm LDA, on a realistic stream of 10 million tweets. Note that, in contrast to the empirical efficiency computation performed in Xie et al. (2013), the complexity here is estimated on theoretical grounds and hence is independent of parameters, parallelization techniques and computing power. We also note

[17] $[1...B]$ in the original paper (Xie et al. 2013).

[18] Table I of (Xie et al. 2013).

[19] http://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations.

[20] In agreement with http://firstmonday.org/ojs/index.php/fm/article/view/4366/3654.

Table 2 Complexity analysis as a function of the corpus dimension

D	t	L	L′	Θ	W	K	$SAX*$	$BiTerm$	$EDCoW$	$TopicSketch$
100	9.4	179	9	3	10	50	7,784	201,348	25,341	16,117,823
1K	9.4	2,401	25	3	10	50	73,086	2,013,480	306,630	47,259,589
10K	9.4	32,155	74	3	10	50	665,382	20,134,800	3,820,434	138,620,347
100K	9.4	430,593	217	3	10	50	6,042,708	201,348,000	48,659,378	407,068,448
1M	9.4	5,766,068	635	3	10	50	55,434,549	2,013,480,000	629,661,338	1,200,080,494
10M	9.4	77,213,473	1.862	3	10	50	**517,658,362**	**20,134,800,000**	**8,238,768,557**	**3,584,819,505**

that BiTerm complexity is negatively influenced by the initial step of bi-term computation, while ECDoW is mainly influenced by the first stage of signal transformation and TopicSketch is penalized by the Topic Inference algorithm. Furthermore, while SAX* and ECDoW are not influenced by the K parameter (the number of clusters), using LDA or TopicSketch on large Twitter streams with growing K becomes prohibitive (as shown when comparing bold values in Table 2). For example, the complexity of LDA with K growing from 50 to 250, according to the experiments described in Yan et al. (2013), increases from 20,134,800,000 to 99,094,800,000. K also impacts on TopicSketch, as shown by the complexity formula: in practice, the authors set K = 5 in their paper, but they do not analyze the effect of this parameter on performance.

4.2 Parameter tuning

This Section is dedicated to parameter tuning. In Sect. 3 we provided intuitive arguments favoring a coarse temporal discretization and a binary alphabet, given the poorly dense nature of our Twitter stream. This Section provides more accurate analysis of the effect of parameters on the quality of detected events.

To support our analysis we used the last two months of our 1 % Twitter stream, from July 1st 2014 to August 30th 2014. This subset includes 276,057,840 tweets, of which 114,797,700 in English. We selected this period because it was very dense in events capturing world-wide attention in the media,[21] namely:

- Two final matches of the World Football Cup (Brazil-Germany and Germany-Argentina)
- The Gaza war
- Malaysia airplane crash in Ukraine
- Ebola outbreak
- The suicide of Robin Williams
- Shooting of Michael Brown and Ferguson protest
- ISIS (Islamic State of Iraq and Syria) killings in Iraq

These events are very diverse in kind (sport, health, disasters, politics, celebrities) and **duration** (some are bursty events, others have plateaus or multiple peaks, as can

[21] This is also confirmed by the fact that we noticed an increment of daily tweets from an average of 3.3M per day during May to 4.6M during August.

Table 3 Parameter settings used in experiments

Parameter	Values
W *(window size)*	2 or 10 days
$\boldsymbol{\Delta}$ (discretization size)	4, 8, 12 and 24 h
$\lvert\boldsymbol{\Sigma}\rvert$ (alphabet dimension)	$\{a, b\}$; $\{a, b, c\}$
f *(frequency threshold)*	3000, 1000, 600, 400, 200
$\boldsymbol{\delta}$ (clustering parameter, see 2.3)	0.25-0.5-0.75

also be monitored on Google Trends). What is more, some of the events overlap, thus giving us the opportunity to test the ability of our SAX* approach, when using different parameter settings, not only to detect all the events accurately, but also to separate them correctly.

Table 3 provides a summary of different parameter values used in the experiment, though for the sake of space we cannot discuss all combinations.

Appendix Table 6 shows the detected clusters in the Summer 2014 corpus, along with the most frequent co-occurring n-grams for some significant subsets of parameter combinations (we show 5 experiments, named A, B, C, D and E). In the table, we indicate the peak of each event as found using Google Trends (GT) and compare it with the cluster peak detected by SAX*. We also underline apparently "spurious" (i.e. belonging to unrelated events on the same day) words in each cluster, if there are any.

At first glance, it would appear that the best selection of parameters is the one we "anticipated" in Sect. 3, i.e. $W = 10, |\Sigma| = 2, \Delta = 24\,\text{h}, f = 1000, \delta = 0.5$ (Experiment A). However, closer inspection reveals some other combination could represent an interesting alternative. In fact, the best choice depends on the features of the available Twitter stream in terms of density and geo-location.

We now discuss the effect of each parameter separately. Firstly, we found that varying parameter δ, the clustering aggregation threshold, has pretty obvious consequences: $\delta = 0.5$ produces the best results with any other combination of parameters, since smaller or larger values have the effect of generating very small clusters or noisy clusters, respectively. Consequently, we do not show the effect of variable δ in Appendix Table 6.

The frequency f and the window W have the following effect: especially in combination with larger W, higher frequency thresholds (Experiments A, B and C) accurately capture world-wide phenomena with a consistent echo on the web, while a smaller f with smaller W (Experiments E) are best suited to capturing local or low-impact phenomena and early or late discussions on larger events (i.e. anticipated or delayed discussions with respect to the Google Trends peak). Consider for example Experiment E: as an example of a low-impact event, we note that this experiment is the only one that captures the Ebola outbreak (which in turn, is the only one which is not captured by Experiment A). In fact, the Ebola outbreak did not obtained a high echo on Twitter[22]: to check this, we also calculated—the frequency of the singleton token "*ebol*".

[22] www.foreignpolicy.com/articles/2014/09/26/why_big_data_missed_the_early_warning_signs_of_ebola.

As an example of anticipated discussion, note that the suicide of Robin Williams actually happened on August 11th, therefore early rumors might have started on the same day (in Experiment E the event is detected on August 11th). But the peak in both GT and our clusters, obtained with larger f (Experiments A, B and C), is on August 12th.

As an example of local clusters, although it is generally difficult to interpret all generated clusters, we list here a number of clusters extracted in Experiment E for which we could find a correspondence when googling on the web:

- Qualification of tennis at Wimbledon on July 5th (*qualifi,gut,tackl,tenni,Wimbledon*)
- Independence day in USA on July 4th (*america, freedom, patriot*)
- Yorkshire country cricket cup on July 5th (*cricket,yorkshir*)
- Liverpool Barcelona football match on July 11th (*Liverpool,Barcelona,lfc*)
- Hamas militants emerging from a *tunnel on August* 1st *(isra, tunnel, premier)*

plus many others, of which some have been omitted for brevity and others because of the difficulty of verifying manually the more than 450 clusters generated by the specified parameter settings of Expriment E.

Finally, we note that clusters with smaller f often mix more events, or include spurious words, like *arsen, vermaelen* in the ISIS clusters (Experiment E). The reason for such "mixed" clusters is that with a small f threshold clusters may include both tokens with a very high, and tokens with a much lower, incidence. Other examples of spurious words are to be found in Experiment D. Experiment D, with $W = 2, \Delta = 8hr, |\Sigma| = 2, f = 600$, represents an intermediate setting: clusters are still good, but more spurious words are found. Note that this happens because we are using a world-wide stream with very variable frequency values of tokens: on a local stream we would expect more homogeneous clusters with small f.

Concerning the alphabet size, we note the following:

- With a binary alphabet (and larger W, Δ, f), all the events are detected except for Ebola (see Experiments A);
- With a binary alphabet the clusters are reasonably pure (i.e. all tokens refer to a single event) and in only two cases in Experiment A does the binary dimension of the alphabet Σ cause a minor event to be clustered with one of the eight main events. The first of these cases concerns the July 10th, 2014 cluster: this is mostly about *Gaza war*, but it also includes two words from a sports event ("alexi, arsen": Alexis Sanchez transferred to Arsenal on July 10th, as we discovered). The second case concerns "shark, moon" in the *Ferguson protest* cluster. On the same day there was a full moon and hence the beginning of the so-called "shark week". As we already discussed, purity worsens with low f and binary alphabet and recall worsens with smaller Δ (Experiment C, D);
- The problem of purity is mitigated in spite of low f, when a larger vocabulary (a, b, c) is used. Clusters seem purer (Experiments B, E), however, recall is lower as nothing at all is found for several events, and furthermore, the dimension of clusters is smaller.

Concerning the discretization Δ, we simply note that, as expected, larger values of Δ are best suited to larger windows W, and vice versa. Experiments with large W and small Δ are not shown in Appendix Table 6 simply because they produce very bad results.

To summarize: large W, Δ and f are best suited to capturing world-wide events; smaller W, Δ and f to detecting local events or early/late topics of a major event. Larger alphabets reduce the dimension and improve purity of clusters, however a binary alphabet in combination with higher W and f performs relatively well, leading to accurate, relatively pure clusters, and a better recall (7 over 8 events are captured). Clearly, the precise notion of "large" and "small" depends on the density of the available Twitter stream. The best parameter values discussed in this Section apply to a poorly dense 1 % Twitter stream. Finally, concerning cluster purity, as will be discussed briefly in the next Section, post processing of tweets related to a detected cluster may offer the possibility of separating out accidentally merged events.

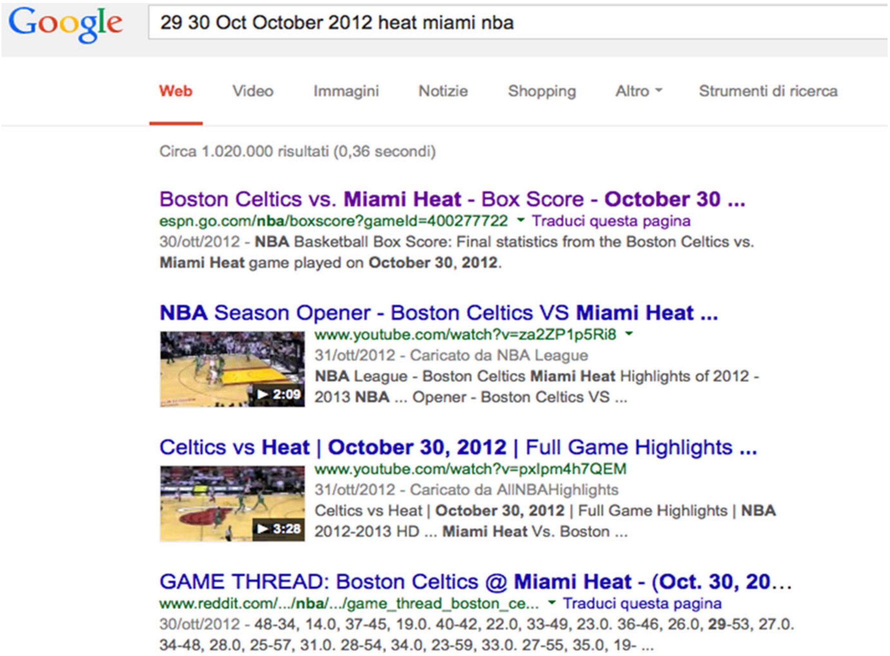
4.3 Qualitative evaluation: event detection

The experiments discussed in the previous Section suggest that the selection of the best combination of parameters ($|\Sigma| = 2, W = 10\,days, \Delta = 24hrs\ f = 1000, \delta = 0.5$), for our 1 % world-wide Twitter stream. In this Section, our aim is to evaluate the precision of SAX* on a 1-year stream, in this case, without any prior knowledge/selection of the events to be captured. However, we used $f = 3000$ rather than 1000, **in order** both to reduce the number of events to be manually labeled in 1 year and to concentrate on more easily verifiable world-wide events.

Overall, in this subset we indexed an average of 69,974,529 terms (L) with frequency higher than f in every window of 10 days, while L' is order of 1000s. After computing the SAX strings in 366 windows we remove all strings not matching the regex (1), a step that greatly reduces the number of words to be clustered. In our previous complexity evaluation (Sect. 4.1) we were very conservative when setting $L' = L''$, since on average the ratio L''/L' in a window W is around 0.06. Finally, we generated the day clusters, as described in Sect. 3.2.

As regards evaluating the quality of retrieved events, our goal was to provide a more objective evaluation than that described in previous literature. To achieve this we match detected events on a day Δ_i against Google snippets in the same day. The evaluation procedure is as follows:

1. For every day cluster, we compute all bigrams, 3-, 4- and 5-g. N-grams are meant to represent the main cluster sub-topics.
2. Using the tweets on the same and previous day, we compute the most frequent maximum n-gram (n = 2, 3, 4, 5). Examples of these n-grams for the Summer 2014 corpus are shown in the last column of Appendix Table 6;
3. We then query the web with this n-gram and the date Δ of the detected event and the previous day, for example: October 29 30 *heat miami nba*;
4. If in the first 10 Google snippets retrieved we find a) a perfect match with the query string, and b) the matching page(s) include a clear description of a related event, we mark the cluster as a hit.



29 30 Oct October 2012 heat miami nba

Circa 1.020.000 risultati (0,36 secondi)

Boston Celtics vs. Miami Heat - Box Score - October 30 ...
espn.go.com/nba/boxscore?gameId=400277722 ▾ Traduci questa pagina
30/ott/2012 - NBA Basketball Box Score: Final statistics from the Boston Celtics vs. Miami Heat game played on October 30, 2012.

NBA Season Opener - Boston Celtics VS Miami Heat ...
www.youtube.com/watch?v=za2ZP1p5Ri8 ▾
31/ott/2012 - Caricato da NBA League
NBA League - Boston Celtics Miami Heat Highlights of 2012 - 2013 NBA ... Opener - Boston Celtics VS ...

Celtics vs Heat | October 30, 2012 | Full Game Highlights ...
www.youtube.com/watch?v=pxlpm4h7QEM
31/ott/2012 - Caricato da AllNBAHighlights
Celtics vs Heat | October 30, 2012 | Full Game Highlights | NBA 2012-2013 HD ... Miami Heat Vs. Boston ...

GAME THREAD: Boston Celtics @ Miami Heat - (Oct. 30, 20...
www.reddit.com/.../nba/.../game_thread_boston_ce... ▾ Traduci questa pagina
30/ott/2012 - 48-34, 14.0, 37-45, 19.0. 40-42, 22.0, 33-49, 23.0. 36-46, 26.0, 29-53, 27.0. 34-48, 28.0, 25-57, 31.0. 28-54, 34.0, 23-59, 33.0. 27-55, 35.0, 19- ...

Fig. 6 Screen dump of Google first hits for the query "29 30 October 2012 heat miami nba"

Note that, unlike the case with the BiTerm LDA model, computing n-grams is for evaluation purposes only and does not affect the complexity of our method. The data presented to the (three) evaluators are organized as in the following example:

Date: October 30th 2012
Cluster: [*arsen heat lebron Miami nba]*
Tweets[23]: Every Celeb basketball fan and Basketball player can't wait 4 Miami Heat 2 open up d NBA and Superstar LeBron is not ustraliad
N-gram: <*heat Miami nba*>
Google:https://www.google.com/search?spell=1\&q=29+30+Oct+October+2012+heat+miami+nba

The evaluation data are made available as supplementary material, along with the retrieved Google page (see Electronic supplementary material).

Interested readers can easily verify that in the great majority of cases the snippets retrieved by Google render the judgment quite straightforward, as is shown in Fig. 6 and confirmed by the high inter-annotator agreement (K-Fleiss is 0.88 in Table 4). Thanks to its simplicity, we believe that this evaluation approach has a clear advantage over manual evaluation of topic clusters, even when such evaluation is performed by an independent team of evaluators. Of course, it would not be possible to evaluate minor

[23] Here we show only one tweet for the sake of space, whereas in our testing dataset we retrieve 10–20 tweets.

Table 4 Precision at detecting events reported on the Web

Total	Positive	Spam	Teen Events	K Fleiss[a]
153[b]	122	10	10	0.88

[a] http://en.wikipedia.org/wiki/Fleiss~%27_kappa
[b] Annotators agreed on 153 events over 159 total analyzed clusters

events with no echo on the web in this way, but with such minor events even a manual evaluation would be very hard to perform.

Table 4 summarizes the results of our evaluation. In Table 4, “Spam” refers to clusters generated by spam messages, while “Teen events” are school-related events not found in Google, such as the beginning of the spring semester, or the English final exam. The total precision is 0.79. When ignoring spam, the precision is 0.85 and when ignoring spam and teen events[24] it is 0.91. Unfortunately, these values cannot be compared with the only two available precision values shown in Table 1 since, on the one hand, the datasets are different, while on the other, the number of evaluators, k-statistics and policy of evaluation adopted by the authors are unknown. However, we can reasonably claim that SAX* is at least state-of-the-art as far as quality of clusters is concerned.

Analyzing the data, errors are caused by the following two factors:

- The accidental merging of two co-occurring events, for example a delay in the release of the new iPhone and an event concerning the artist Frank Ocean, as in: *delai iphon australia award ocean*. Two other examples_have already been discussed in Sect. 4.2, but only in one case (*Gaza war* in Experiment A, see Appendix Table 6) was the “wrong” n-gram extracted. A Twitter stream with higher coverage wrt the available 1 % would allow a more fine-grained analysis in order to separate the events (for example, adding geo-localization). Another solution would be to separate accidentally merged events in a post-processing step based on the co-occurrence of most frequent n-grams,. For example, tweets with “*delai iphone*” would not include any of the other n-grams in the cluster, such as “*australia award ocean*”. When creating a connection graph of the detected cluster n-grams, separated events produce disconnected graph components We leave this extension to future research.
- A second source of error is caused by stemming, for example the three-gram “dalei dave tom” on August 11th 2012 refers to Tom Daley Olympic diving final, but the above three-gram does not match the corresponding event on Google (however, it does match with “daley dave tom”). As a matter of fact, considering such cases as errors, or not as errors, was the only matter of disagreement between the two evaluators. This is confirmed by the high value of the k-Fleiss measure. In any case, removing the stemmer would produce worse results because of data sparseness. It is likely that a Twitter stream larger than 1 % could produce good results even without stemming.

[24] These events can easily be filtered out by a classifier, however teen events could be of interest.

Finally, even though it is not clear how to compute recall, an analysis of Wikipedia Events shows that we identify all the major events (except for an accidental data hole during the Obama election days) that had an echo on the web: the European football championships, the important debates between Romney and Obama, US Independence day, the deaths of Michael Duncan and Margaret Thatcher, Hurricane Sandy, Thanksgiving and Christmas, the Boston Bomb, the new Pope and a variety of sports, gossip and entertainment events. Considering, instead, the shorter "Summer 2014" corpus used in Sect. 4.2, the recall of the best parameter combination was 7/8 events, even though, as we noted above, clusters may in some cases include words of other co-occurring minor events. Note that, as previously discussed, co-occurring world-wide events in general are well separated.

In order to assess the cluster quality of alternative approaches, we considered BiTerm LDA (Yan et al. 2013). As we already remarked, we cannot compare with TopicSketch and ECDoW since neither a common dataset nor an implementation of these methods is publicly available. On the other hand, even though a comparison with a topic-based method may not seem entirely appropriate, it is also_worth repeating what we mentioned in the Introduction, i.e., that a large body of event detection research based on LDA exists.

We compared the two methods over a 10-day window starting on February 2nd, 2013. We selected this particular temporal window since it includes two events with a large echo: the end of Super Bowl on February 4th, and the Grammy Awards on February 11th. For LDA, we used the implementation available from *sourceforge*,[25] while for BiTerm LDA we used the implementation provided by the authors[26] Words were stemmed and we removed, as we also did for SAX*, stems with less than three characters and less than 3000 occurrences. Since in Yan et al. (2013) no additional filters based on the temporal behavior of terms are applied, the survived vocabulary is L'. LDA requires the number of clusters to be fixed, and we set this number to 20 (we found that higher values produce worse results). Finally we used the same "objective" procedure for evaluating SAX*, and BiTerm LDA: we considered the set of related Tweets for each cluster, we computed the most frequent maximum cluster-related n-gram, with n=2,3,4 or 5, and then we looked for a match on Google. With SAX*, we detected 7 daily events in the window, of which, only 1 was wrong (according to our evaluation procedure). Among the other events, we correctly detected the SuperBowl and Grammy Awards, e.g.:

On 4 Feb 2013 00:00:00 GMT:

<bowl,lewi,rai,raven,super> from: [*alex, raven, rice, ring, root, san, super, superbowl, touchdown, yard,baltimor, bowl,commerce,field, footbal, halftim, lewi, murder, nfl, niner, puppi, quarter, rai, rais*]

On 11 Feb 2013 00:00:00 GMT:

<bob,bruno,gramm,mar,marlei> from: [*Miguel mumford, ocean, perri, princ, rihanna, swift, taylor, timberlak, wiz, adam, adel, artist, bob, bruno, carri, frank, hunter, kati, kelli, mar, marlei, everywhere, award, dress, grammi, justin*]

[25] http://jgibblda.sourceforge.net/.

[26] http://code.google.com/p/btm/.

Table 5 Summary statistics of SAX* and BiTerm LDA during a 10-day window starting on February 2nd, 2013

	Processing time (s)	Average # of terms per cluster	Total number of different terms in clusters	Number of clusters
SAX*	231.5612	14.6953	88	**7**
BiTerm	42,498.21381	15	120	20

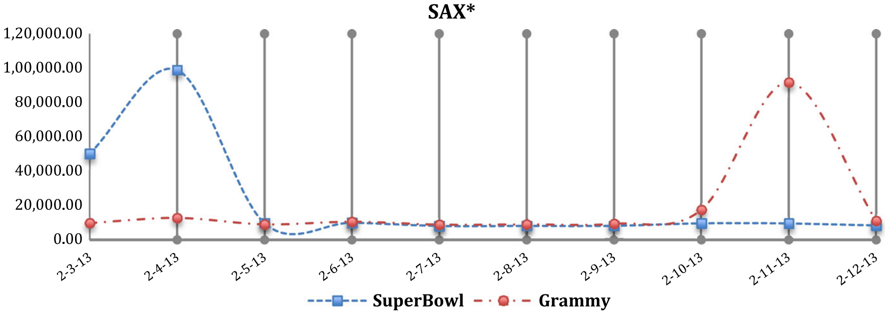


Fig. 7 Time line for Grammy Award and Superbowl events with SAX*

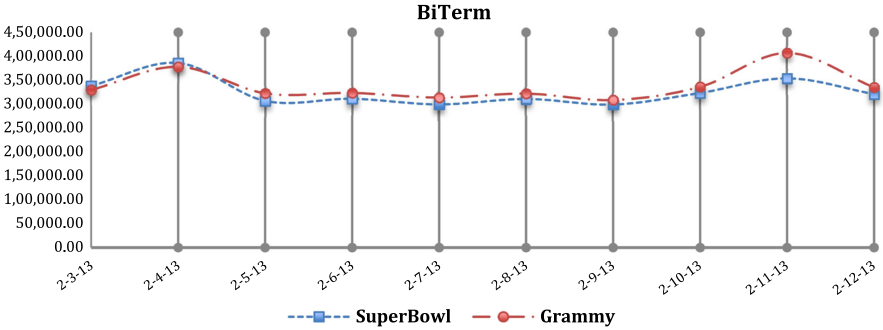


Fig. 8 Time line for Grammy Award and Superbowl events with BiTerm LDA

With BiTerm LDA, out of 20 clusters, the only "positive" cluster is: < *bowl super who win you* > from [*game super win bowl plai watch team raven all you superbowl just who get now*]

This experiment is also made available in the previously indicated link.

Table 5 shows a summary statistics of this experiment. The experiment shows once again that our approach outperforms the others in terms of complexity and quality of the detected events. Note in Table 5 that the actual required processing time was about 231 secs for SAX*, and 42,498 secs for BiTerm LDA, which is a remarkable difference.

Furthermore, we computed the Jaccard similarity between the two clustering results, which is 0.05583 (there are 11 terms in common between the two sets of clusters, and $L'_{BiTerm} \cup L'_{SAX*} = 197$).

As a final note, we would like to illustrate by means of an intuitive example the difference between approaches that look for textual similarity (as do LDA and many others) and those based, as ours is, on temporal similarity. Figure 7 shows the time line of SAX* Grammy Awards and SuperBowl clusters (in the Figure, the time line of a cluster is generated with the "OR" of all its terms). Figure 8 shows the time line for BiTerm LDA, obtained using the same procedure, for the matching SuperBowl cluster and for the best manually-selected Grammy Award cluster (as previously remarked, the ngram-based automated procedure does not identify a Grammy Award cluster). Comparing the two Figures helps to clarify the main difference between the two approaches: SAX* maximizes term cohesion in *time*, while topic models maximize term cohesion in *documents*. While both these are reasonable optimization objectives, the former is clearly more appropriate for the task of event discovery.

5 Concluding remarks

In this paper we presented a novel micro-blog event discovery algorithm, named SAX*, based on the notion of temporal, rather than contextual, co-occurrence. Temporal co-occurrence overcomes the problem arising from the very limited context provided by users in their messages. We demonstrated that SAX* is able to detect and trace patterns of collective attention in a very precise way, and in addition, that its complexity is one or two orders of magnitude lower than previous methods presented in literature. Compared to previous work in this area, our contribution has several advantages:

1. First, we demonstrate that our SAX* methodology is much less computationally intensive (by orders of magnitude) than previous approaches to event detection based on latent topic models, or other temporal clustering approaches. This makes our method applicable to lengthy micro-blog streams or lengthy texts in general, while at the same time also being capable of capturing precisely the latent topics in short messages. Thanks to its computational efficiency we were able to experiment our method on a full year of Twitter traffic (several terabytes of data) rather than on just a few weeks of data, as is the case for the majority of published evaluations.
2. Second, we provide a suitable general characterization for the temporal behavior of events that is more accurate and general than a simple "bursty" model. Notably, by using just a a single regex our generalized model is able to capture the general patterns of attention described in other papers on this topic (Lehmann et al. 2012; Yang and Leskovec 2011);
3. Third, we provide a detailed analysis of the effect of parameter settings on the quality of detected events. Even in the absence of a locally dense Twitter stream for additional tests, we could still infer the relations between parameter setting, features of the micro-blog stream to be analyzed, and performance;
4. Fourth, we propose a more "objective" and much less labor-intensive manual evaluation strategy for event detection. This is based on using the detected event clusters and date to create a web query to be used in searching for a match. Our results are made available, as supplementary material, for comparison and easy replication of the experiments;

5. Fifth, in order to allow subsequent comparison with other models, we will also make available[27] (under the restrictions imposed by Twitter) the Summer 2014 stream. This stream has several advantages over the Singapore users stream, since it includes several well-known world-wide events of diverse kinds (sports, disasters, health, politics, media), thus making evaluation easier for researchers.
6. Finally, we note that, apart from stemming, our approach is language independent—which is useful even if not crucial.

There are also a number of limitations, which we hope to reduce in our future work:

1. The dataset that we used is poorly dense and we have not performed a geo-localization of tweets, as a consequence our hunch that a smaller temporal grain (hours or minutes) is more appropriate for identifying local phenomena could not be adequately tested in this paper.
2. Since in SAX* words are clustered on the basis of temporal co-occurrence and shape similarity, rather than co-occurrence in tweets, co-occurring events may get accidentally merged in the same cluster. As shown in the examples (e.g. the clusters in Appendix Table 6), this is relatively rare when parameters are tuned to detecting big world-wide, events, but is more frequent when the frequency threshold is lowered. In our future work we aim to reduce this problem in a post-processing step. In fact, we have noticed that in the co-occurrence graph of detected clusters, disconnected components identify unrelated events.

Appendix

See Table 6.

Table 6 "Summer 2014" experiments

Experiment parameters	Events	Peak date on Google Trends	Most frequent detected cluster peaking on date	Date of cluster peak	Best tweets n-gram on peak
Exp A : $\mathbf{W = 10, \Sigma = \{a, b\}\ \Delta = 24h, f = 1000}$	World football cup Germany-Brazil (G-B)	July 8th	Ger,germani, half, isso, ronaldo, stadium,tem, brasil,brazilian, match,nurs, predict, puta,qualif i,score, semi, vamo, worldcup, yellow, chile, countri, cup, dive,european, fifa, football,foul,game,injur, keeper,kick	July 8th	David,fred, hulk, luiz,oscar

[27] Requests must be addressed to the authors.

Table 6 continued

Experiment parameters	Events	Peak date on Google Trends	Most frequent detected cluster peaking on date	Date of cluster peak	Best tweets n-gram on peak
	Gaza war (GW)	July 10th	Reveng,weav, gaza, Israel, jew, palestin, spain,alexi, arsen,border,budget	July 10th	Alexi,arsen**
	World football cup Germany-Argentina (G-A)	July 13th	Gol,isso,net,pitch,tem, third, yellow,defenc, field,game, Germani,half,lose,nazi, predict, ronaldo, score,soccer, stadium,sweepstak, win,winner,cup, defeat,deserv,fifa, final, football	July 13th	Cup,deserv, Germani,win
	Malaysia airplane crash in Ukraine (Ukr)	July 17th	Passeng,plane,purg, Russian,Ukrain,airlin, crash,flight,kik, Malaysia,Malaysian	July 17th	Airlin, crash, flight, Malaysia, Ukrain
	Ebola outbreak (EB)	August 8th	–	–	–
	Shooting of Michael Brown in Ferguson and protest (Ferg)	August 14th	Citizen,cnn,investig, mike,militari,situat, speech,brown,brutal	August 12th	Brown, mike
	Robin Williams suicide (RW)	August 12th	Brunomar,mimi,sad, sharkweek,societi, suicid, transform, william,actor, captain,childhood,comedi, comedian,depress, doubtfir, fri,hook,hunt, jumanji, laughter, legend, loss,pan,peac, poet,riot,rip, robin	August 12th	Actor, comedian, rip,robin, william
	Northen Iraq Offensive (ISIS)	August 20th	Terrorist,balotelli, isi,saudi,terror	August 21st	Isi, Saudi, terror

Table 6 continued

Experiment parameters	Events	Peak date on Google Trends	Most frequent detected cluster peaking on date	Date of cluster peak	Best tweets n-gram on peak
Exp B : $\mathbf{W = 10}$, $\boldsymbol{\Sigma = \{a, b, c\}}$ $\boldsymbol{\Delta = 24h}$, $\boldsymbol{f = 1000}$	G-B	July 8th	Fifa,worldcup,score, defeat,defend, humili, nazi,ronaldo	July 8th	Score,worldcup
	GW	July 10th	Isra,Israel,palestin, gaza,hama	July 10th	Gaza,Israel
	G-A	July 13th	Argentina,Germani, ozil, ronaldo,predict	July 13th	Argentina, Germani, ronaldo
	Ukr	July 17th			
	EB	August 8th			
	Ferg	August 14th	Ferguson,cop,polic, protest,racism	August 14th	Ferguson, polic, protest
	RW	August 12th	Childhood,doubtfir, jumanji, robin,suicid	August 12th	Childhood, doubtfir, jumanji, robin
	ISIS	August 20th			
Exp C : $\mathbf{W = 10}$, $\boldsymbol{\Sigma = \{a, b\}}$ $\boldsymbol{\Delta = 12h}$, $\boldsymbol{f = 3000}$	G-B	July 8th	Bra,Brazil,hulk, luiz,neymar,oscar, silva,thiago	July 8th	Brazil,neymar, silva,thiago
	GW	July 8th			
	G-A	July 10th	Alli,suppos,gotz, messi,soccer, champion, Germani	July 13th	Germani,messi
	Ukr	July 13th	Ukrain,thumb,passeng, plane,airlin, Malaysia, Malaysian, missil	July 17th	Airlin, Malaysian, passeng, Ukrain
	EB	July 17th			
	Ferg	August 8th			
	RW	August 14th	Fri,william,comedi, hunt,laughter,peac, commit,robin,sad, suicid,actor, childhood, comedian,depress,doubtfir, hook,jumanji,loss, poet,rip	August 12th	Actor, comedian,rip, robin,william
	ISIS	August 8th			

Table 6 continued

Experiment parameters	Events	Peak date on Google Trends	Most frequent detected cluster peaking on date	Date of cluster peak	Best tweets n-gram on peak
$\mathbf{Exp\,D}:\mathbf{W}=\mathbf{2}$, $\boldsymbol{\Sigma}=\{\boldsymbol{a},\boldsymbol{b}\}$ $\boldsymbol{\Delta}=\mathbf{8}\boldsymbol{h}$, $\boldsymbol{f}=\mathbf{600}$	G-B	July 8th	Anywher,pll,wale, bra,brager,Brasil, bravsger,Brazil, Brazilian, comeback,David,embarrass, fred,ger,germani, hitler,hulk,humili, klose,luiz,muller, neuer,neymar,oscar, ozil,rape,record, refus,releas,riot, silva,thiago,unbeliev	July 8th	David,fred, hulk,luiz, oscar
	GA	July 10th	Gaza,palestin, isra,Israel,hama	July 10th	Gaza,Israel
	G-A	July 13th	Ger,gerarg,germani, gervsarg,golden,gotz, higuain,jade,klose, lift,littlemix,mario, messi,muller,neuer, offsid, predict,root, soccer,somewher,trophi, win,winner,won, arg,argentina,bbc, boot,champion,christ, congrat,congratul,cup, del,deserv,		Argentina,cup, fifa,final, Germani
	Ukr	July 17th	Russia,Ukrain,vma, airlin,crash,flight, Madrid,Malaysia, Malaysian, mtv,nomin, passeng, plane	July 17th	Airlin, crash, Malaysian, passeng, Ukrain
	EB	August 8th			
	Ferg	August 14th	Report,riot,arrest, congratul,cop,decis, ferguson,govern,justic, polic,presid,protest	August 14th	Ferguson,polic
	RW	August 12th	Becaus,depress,laughter, legend,loss,peter, rip,robin,sad, shock,societi,suicid, talent,william,actor, aladdin,childhood,comedi, comedian,dead,doubtfir, flubber,goodby,grew, hook,hunt,jumanji, peac,poet,rest	August 12th	Dead,poet, robin,societi, william
	ISIS	August 20th			

Table 6 continued

Experiment parameters	Events	Peak date on Google Trends	Most frequent detected cluster peaking on date	Date of cluster peak	Best tweets n-gram on peak
$\mathbf{Exp\,E}:\mathbf{W}=\mathbf{2}$, $\boldsymbol{\Sigma}=\{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\}$ $\boldsymbol{\Delta}=\mathbf{4}\boldsymbol{h}$, $\boldsymbol{f}=\mathbf{200}$	G-B	July 8th			
	GA	July 10th	gaza,tragedi	July 17th	Gaza,tragedi
	G-A	July 13th			
	Ukr	July 17th	Ukrain,Malaysia, passeng	July 17yh	Malaysia, ukrain
	EB	August 8th	Viru,ebola, reduc	August 31st	Ebola,viru
	Ferg	August 14th			
	RW	August 12th	Flood,suicid,tragic, william,aladdin, comedian, depress, devast,doubtfir, flubber,jumanji, legend, poet,robin	August 11th	Doubtfir, flubber,jumanji, robin,william
	ISIS	August 20th	Kany,promis,nigeria, india,woke,swifti, eau,launch,oper, bound,isi, iraq,sunni,bbc, arsen,vermaelen	August 7	Iraq,isi

References

Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J. Mach. Learn. Res. 3:993–1022

Chae J, Thom D, Bosch H, Jang Y, Maciejewski R, Ebert D, Ertl T (2013) Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition. IEEE symposium on visual analytics science and technology, Seattle

Cha M, Haddadi H, Benvenuto F, Gummadi K (2010) Measuring user influence in twitter: the million followers fallacy. In: Proceedings of conference on artificial intelligence AAAI

Cheng T, Wicks T (2014) Event detection using Twitter: a spatio-temporal approach. PLoS One 9(6):e97807. doi:10.1371/journal.pone.0097807

Dao Q, Jiang J, Zhu F, Lim WP (2012) Finding bursty topics from microblogs. In: Proceedings of conference association of computational linguistics ACL 2012

Dou W, Wang X, Ribarsky W, Zhou M (2012) Event detection in social media data. In: IEEE VisWeek workshop on interactive visual text analytics. Seattle, WA

Dredze M (2012) How social media will change public health. IEEE Intell Syst 27(4):81–84. doi:10.1109/MIS.2012.76

Hong L, Davison B (2010) Empirical study of topic modeling in twitter. In: Proceedings of the first workshop on social media analytics, pp. 80–88. ACM

Hong L, Dom B, Gurumurthy S, Tsioutsioulikis K (2011) Time-dependent topic model for multiple text streams. In: ACM conference on knowledge discovery and data mining KDD 2011, San Diego

Huang B, Yang Y, Mahmood A, Wang H (2012) Microblog topic detection based on LDA model and single-pass clustering RSCTC 2012, LNAI 7413, pp. 166–171

Ifrim G, Shi B, Brigadir I (2014) Event detection in Twitter using aggressive filtering and hierarchical tweet clustering proceedings of SNOW-WWW workshop, Korea

Jain A (2010) Data clustering: 50 years beyond K-means. Patt Recogn Lett 31:651–666

Keogh E, Chakrabarti K, Pazzani M (2001) Locally adaptive dimensionality reduction for indexing large time series databases. In: Proceedings Of ACM special interest group on management of data SIGMOD, pp. 151–162

Kovacs F, Legany C, Babos A (2005) Cluster validity measurement techniques. In: Proceedings of 6th international symposium of Hungarian researchers on computational intelligence, Budapest

Lee R, Sumiya K (2010) Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection. Proceedings of the 2nd ACM international workshop on location based social networks SIGSPATIAL, LBSN ’10. ACM, New York, pp. 1–10

Lehmann J, Goncalves B, Ramasco JJ, Cattuto C (2012) Dynamical classes of collective attention in Twitter. Proceedings of World Wide Web Conference WWW2012

Lin J, Keogh E, Li W, Lonardi S (2007) Experiencing SAX: a novel symbolic representation of time series. Data Mining Knowl Discov 15(2):107–144

Lin J, Khade R, Li Y (2012) Rotation-invariant similarity in time series using bag-of-patterns representation. J Intell Inf Syst 39:287–315

Li C, Sun A, Datta A (2012) Twevent: segment-based event detection from tweets. In: Proceedings of ACM international conference on information and knowledge management CIKM

Maynard D, Funk A (2012) Challenges in developing opinion mining tools for social media. In: Proceedings Of @NLP cann u tag #usergenartedcontent? Workshop at LREC 2012, Istanbul

McMinn A, Moshfeghi Y, Jose JM (2013) Building a large scale corpus for evaluating event detection in twitter, ACM international conference on information and knowledge management CIKM’13, San Francisco

Mei Q, Zhai C (2005) Discovering evolutionary theme patterns from text—an exploration of temporal text mining. In: Proceedings of conference of knowledge discovery and data mining KDD’05, Chigago

Oncina J, Garcıa P (1992) Inferring regular languages in polynomial updated time. In: 4th Spanish symposium on pattern recognition and image analysis, MPAI. vol. 1. World Scientific, pp. 49–61

Petrovic S, Osborne M, Lavrenko V (2010) Streaming first story detection with application to Twitter. In: Proceedings of national American conference of the association of computational linguistics NAACL

Petrovic S, Osborne M, Mc Creadie R (2013) Can Twitter replace Newswire for breaking news?. In: Proceedings of the 7th international AAAI conference on weblogs and social media, ICWSM

Pohl D, Bouchachia A, Hellwagner H (2012) Automatic sub-event detection in Emergency management using social media (2012), WWW2012-SWDM’12 Workshop, Lyon

Popescu AM, Pennacchiotti M, Paranjpe D (2011) Extracting events and event descriptions from twitter. In: Worls Wide Web Conference WWW2011, pp. 105–106, 2011

Rui L, Kin L, Ravi K, Kevin C (2012) TEDAS: a Twitter-based event detection and analysis system. In: IEEE 28th international conference on data engineering (ICDE), pp. 1273–1276

Wang X, Zhu F, Jing J, Li S (2013) Real time event detection in Twitter, conference on web age information management WAIM, Spinger

Weng J, Lim E, Jiang J, He Q (2010) Twitterrank: finding topic-sensitive influential twitterers. In: Proceedings of the third ACM international conference on Web Search and data mining WSDM, ACM, pp. 261–270

Weng J, Yao Y, Leonardi E, Lee B (2011) Event detection in Twitter. In: International AAAI conference on weblogs and social media ICWSM

Xie W, Zhu F, Jang J, Lim E, Wang K (2013) TopicSketch: real-time bursty topic detection from Twitter, IEEE 13th international conference on data mining (ICDM)

Yang J, Leskovec J (2011) Patterns of temporal variation in online media. In Proceedings of the fourth ACM international conference on web search and data mining (WSDM), pp. 177–186

Yan X, Guo J, Lan Y, Cheng X (2013) A biterm topic model for short texts. In: World Wide Web conference WWW 2013, Rio de Janeiro