Web Intelligence 1 (2018) 1–5
IOS Press

The Importance of Unexpectedness: Discovering Buzzing Stories in Anomalous Temporal Graphs

Francesco Bonchi [a], Ilaria Bordino [b], Francesco Gullo [b] and Giovanni Stilo [c]
[a] *ISI Foundation, Italy*
E-mail: francesco.bonchi@isi.it
[b] *UniCredit, R&D Department, Italy*
E-mail: ibordino@acm.org, gullof@acm.org
[c] *Sapienza University, Italy*
E-mail: stilo@di.uniroma1.it

Abstract. The real-time nature and massive volume of social-media data has converted news portals and micro-blogging platforms into social sensors, causing a flourishing of research on story or event detection in online user-generated content and social-media text streams. Existing approaches to story identification broadly fall into two categories. Approaches in the first category extract stories as cohesive substructures in a graph representing the strength of association between terms. The latter category includes approaches that analyze the temporal evolution of individual terms and identify stories by grouping terms with similar anomalous temporal behavior.

Both categories have their own limitations. Approaches in the first category are unable to distinguish ever-popular concepts from stories that *buzz* in a time interval of interest, i.e., attract an amount of attention that deviates significantly from the typical level observed. The second category ignores term co-associations and the wealth of information captured by them.

In this work we advance the literature on story identification by profitably combining the peculiarities of the two main state-of-the-art approaches. We propose a novel method that characterizes abnormal association between terms in a certain time window and leverages the graph structure induced by such anomalous associations so as to identify stories as subsets of terms that are cohesively associated in this graph. Experiments performed on two datasets extracted from a real-world web-search query log and a news corpus, respectively, attest the superiority of the proposed method over the two main existing story-identification approaches.

Keywords: Story identification, event detection, temporal graphs, dense-subgraph extraction, anomaly detection, web-search logs

1. Introduction

The gargantuan growth of social media has opened a goldmine of data about events taking place around the world. The real-time nature and massive volume of this data has converted news portals and micro-blogging platforms into social sensors, which people increasingly turn to for breaking news and directions about emerging events, often more timely and more effectively than official communication channels. The aforementioned reasons have caused a flourishing of research on event detection in social-media text streams, and event detection has become a well-studied task in information retrieval and data mining [1,14,52,66,92].

Research in recent years has especially uncovered the increasingly important role of leveraging social-network data in disaster situations [21], showing its crucial potential to enhance situational awareness during crisis situations [4,39,38,63,87], such as natural disasters, large-scale malfunctioning events, or terror attacks. Automatically detecting and categorizing *un-*

expected and *anomalous* events in a timely and efficient manner can provide valuable information to support first responders, public-safety agents, as well as local, national and international organizations. Researchers have recently proposed several visual analytics approaches aiming at real-time microblog analysis, providing means to identify anomalous events and to uncover valuable information that can be spread in the aftermath of disasters [40,51,57,5].

Due to its crucial importance for such real-world applications, the problem of automatically identifying stories or events[1] from online user-generated content has recently attracted a great deal of attention in the research community [8,12,26,74,80,92,98,69]. Generally speaking, the goal is to take data from online sources, such as queries issued to a web search engine, news articles, or posts from micro-blogging/social-networking platforms, and automatically extract sets of terms or entities that provide a good description of relevant events happening in the real world.

Approaches to story identification can be classified into two categories. Approaches in the first category identify stories by building a graph representing the strength of association between terms (or entities), and then looking for sets of terms (subgraphs) that are cohesively connected in the graph according to a certain notion of cohesiveness [1,12,24,26,71, 73,74,92,95,98]. The degree of association between any two terms, i.e., the weight assigned to each edge in the co-association graph, is established by counting how many times those terms co-occur in the specific dataset considered (e.g., how many web-search queries, tweets, or posts contain both terms), or by means of correlation measures (e.g., log-likelihood ratio, correlation coefficient) computed on top of the raw co-occurrence counting. Because the strength of association between terms changes over time, the co-association graph actually corresponds to a time-evolving graph, composed of various (deterministic) snapshot graphs. Each snapshot models the co-associations observed at a specific time instant. As an example, if a daily granularity is adopted, each snapshot may represent the number of times any two terms co-occur in a query, news, tweet, or post generated in that day. A major limitation of these approaches is that cohesive subgraphs corresponding to stories are extracted on the snapshot graph observed at the current time instant, that is without considering how the associations between terms have evolved over time or deviated from normality.

The second category of story-identification approaches includes methods that focus on the temporal evolution of the occurrences of individual terms [32, 41,46,54,80,81,88,96]. Such methods assign each term a time series, describing how anomalous (according to a specific anomaly-detection model) its level of occurrence at any time instant is, when compared to the normal level of the whole time horizon. These approaches do not exploit any co-association graph, that is, they do not examine how terms are related to each other and how such relations change over time. Stories are rather identified by analyzing each term individually, and a-posteriori grouping terms based on the similarity of the corresponding anomaly time series. Associations between terms constitute a paramount source of information, which provides valuable insights for assessing to which extent the terms in a story are correlated to each other.

In this work we propose a novel method for identifying stories from user-generated content, which overcomes the limitations of the two main aforementioned approaches by taking both term co-associations and their (anomalous) temporal evolution into account. The proposed method consists of two steps: (i) applying an anomaly model to quantify how abnormal the association between two terms is at any time, with respect to its history, and (ii) leveraging the graph structure induced by such anomalous associations to identify cohesive subsets of terms that are strongly and anomalously associated with each other in a given time window. Our method identifies what we call *buzzing stories*, i.e., stories described by sets of terms that are strongly associated to each other and, at the same time, raise an exceptionally-high level of attention in the time window considered, compared to what normally observed. The main conceptual, technical and empirical contributions of this work are as follows:

- We advance the state of the art on story identification by devising a novel method that addresses the limitations of the main existing approaches.
- The first step of our method assigns, for any time instant, an anomaly score to each pair of terms, so as to reflect the anomaly of the association between those terms at that specific time. To this end, we devise an anomaly-detection model for temporal data that trades off between simplicity, efficiency, and effectiveness .

[1] We use "story" and "event" interchangeably through the paper.

- The second step extracts cohesive subgraphs from the graph induced by the anomalous term co-associations derived in the first step. We define a notion of temporal density to be exploited for the identification of subgraphs that are cohesively connected within a time window of interest.
- We formulate a combinatorial-optimization problem aimed at maximizing the proposed temporal density notion. We theoretically characterize the problem by proving its **NP**-hardness and showing how a relaxation of it has interesting connections with the well-established problem of finding the inner-most core of a graph.
- Inspired by the latter connection, we design an algorithm that approximates our original NP-hard problem effectively and efficiently.
- We perform an extensive evaluation on two real-world datasets, which were extracted from the query log of a popular search engine and a news corpus collected from a number of major Italian newspapers, respectively. Results on both datasets confirm that the proposed method outperforms the two main existing story-identification methods in detecting stories that both raise an anomalous level of attention and match real-world events.

The rest of the paper is organized as follows. Section 2 formalizes the problem of story identification from user-generated content and presents the novel two-step approach proposed in this work. In Sections 3 and 4 we report our experimental evaluation on search-log data and news data, respectively. Section 5 overviews the related literature, while Section 6 concludes the paper.

2. Anomalous Temporal Subgraph Discovery

We are given a set of objects $\mathcal{O}$, a discrete time horizon $\mathcal{T}$, and a function $f : \mathcal{O} \times \mathcal{O} \times \mathcal{T} \rightarrow \mathbb{R}^+$ that, for every time instant in $\mathcal{T}$, assigns a positive real value to every (unordered) pair of objects in $\mathcal{O}$.

$\mathcal{O}$ keeps track of all objects used to describe stories. Objects may correspond to terms or entities extracted from a source of user-generated content, such as posts from micro-blogging or social-networking platforms, news articles, or web-search logs [8,92,98]. $\mathcal{T}$ represents the overall time horizon where the objects in $\mathcal{O}$ are assumed to "interact" with each other. Specifically, $\mathcal{T}$ corresponds to a finite set of time instants, where every time instant $t \in \mathcal{T}$ identifies a basic unit of time within the overall time frame, e.g., an hour, a day, or a week. Function f quantifies the strength of association between two objects in $\mathcal{O}$ at any time instant in $\mathcal{T}$. As an example, for any two objects $o_1, o_2 \in \mathcal{O}$ and a time instant $t \in \mathcal{T}$, $f(o_1, o_2, t)$ can be defined as the number of times o_1 and o_2 co-occur in the data snapshot captured at time t, as well as the log-likelihood ratio or correlation coefficient computed on top of the raw co-occurrence counting [12,74].

We can alternatively think of the input above as a time-evolving (or temporal) undirected weighted graph $\mathcal{G} = (V, \{E_t, f_t\}_{t \in \mathcal{T}})$, i.e., a graph with vertex set $V = \mathcal{O}$, and edge set that varies over time. In particular, every time instant $t \in \mathcal{T}$ is assigned an edge set $E_t = \{\{u, v\} \in 2^V \mid f(u, v, t) \geq \eta\}$, and a function $f_t : E_t \rightarrow \mathbb{R}^+$ assigning weights to edges in E_t in such a way that $f_t(u, v) = f(u, v, t)$. η is a threshold denoting when the strength of association between two objects can safely be assumed to be null, or, equivalently, when the edge between those objects at the corresponding time instant t can be discarded. η is set depending on the application context. Given a temporal graph $\mathcal{G} = (V, \{E_t, f_t\}_{t \in \mathcal{T}})$ and a time instant $t \in \mathcal{T}$, we denote by $deg(u, t)$ the (weighted) degree of vertex u at time instant t, i.e., $deg(u, t) = \sum_{(u,v) \in E_t} f_t(u, v)$. Similarly, given a subgraph of $\mathcal{G}$ induced by a subset of vertices $S \subseteq V$, we denote by $deg_S(u, t)$ the degree of vertex u at time t in that subgraph, i.e., $deg_S(u, t) = \sum_{(u,v) \in E_t, v \in S} f_t(u, v)$. For the sake of simplicity, we slightly abuse of notation and hereinafter denote by S both a subset of vertices of $\mathcal{G}$ and the corresponding subgraph induced by S. Note also that, for ease of notation, we assume the vertex set of each snapshot in the temporal graph fixed. In practice, all singleton vertices of a snapshot can actually be discarded. Thus, the vertex set of each snapshot actually contains only those vertices that have non-zero degree in that snapshot. This way, a temporal graph can easily model situations where a vertex (dis)appears over time. Indeed, without loss of generality, one can assume that any vertex u appearing for the first time at time instant t_i is still contained *as a singleton vertex* in the vertex set of the snapshots at time $t \in [t_0, t_{i-1}]$. Analogously, if a vertex disappears from a temporal snapshot, it can be considered to still be part of the vertex set of that snapshot, but, again, as a singleton vertex.

In this work we study the problem of identifying *buzzing stories* from user-generated content.[2] We assume the input data to be represented by means of a temporal graph $\mathcal{G}$, as described above. Given a temporal graph $\mathcal{G}$ and a time window $W \subseteq \mathcal{T}$, our aim is to extract K stories or subsets of objects that exhibit an *anomalous* behavior in the window W. Here "anomalous" means that the strength of association between the objects forming a story diverges substantially, in every time instant belonging to the window W, from the typical level observed throughout the whole horizon $\mathcal{T}$.

To accomplish our goal we devise a two-step approach. The former step consists in deriving an *anomalous temporal graph* $\mathcal{G}^A$ from the input graph $\mathcal{G}$. $\mathcal{G}^A$ is a graph whose structure corresponds to the structure of $\mathcal{G}$, i.e., vertex and edge set remain the same. What changes is the scoring functions assigning weights to edges. The original functions $\{f_t\}_{t\in\mathcal{T}}$, which weigh edges in $\mathcal{G}$ based on the raw association scores between the corresponding objects, are replaced with functions $\{\phi_t\}_{t\in\mathcal{T}}$ that assign edge weights in $\mathcal{G}^A$ in terms of *anomaly scores*: each score $\phi_t(u,v)$ indicates how anomalous the association between objects u and v is at time instant t with respect to the typical association observed during the entire time period $\mathcal{T}$. The second step takes the anomalous temporal graph $\mathcal{G}^A$ and a time window $W \subseteq \mathcal{T}$ as input, and extracts subsets of objects that are strongly associated to each other in W. This is achieved by looking for subgraphs of $\mathcal{G}^A$ that are *cohesive* enough according to a notion of cohesiveness, which is defined based on the anomaly scores and the given time window.

In the remainder of this section we discuss both steps in detail. Sections 2.1 and 2.2 respectively describe the method to compute the anomaly scoring functions $\{\phi_t\}_{t\in\mathcal{T}}$, and the extraction of cohesive anomalous subgraphs representing buzzing stories, while Section 2.3 summarizes the overall proposed approach.

2.1. Step 1: Computing anomaly scores

The first step of our approach corresponds to a task of anomaly detection in temporal data: assign a score to every data point of a temporal sequence according to a model that quantifies its level of anomaly with respect to the remaining points [33]. In our context we have a temporal sequence for each edge in the input graph $\mathcal{G}$, and the data points in each sequence correspond to the (raw) weights assigned to the corresponding edge over all time instants.

In the following we describe the specific anomaly-detection model employed in this work (other models can be used). This is a model that trades off between simplicity, efficiency, and effectiveness, and gives high-quality results in practice, as testified by our evaluation in Section 3. Our approach is however parametric to the anomaly-detection model: any other existing model can be used.

We rely on an unsupervised approach that first assigns to each edge e at time t a score designed to reflect the relative importance of its weight $f_{t_i}(e)$ with respect to all other edges at time t. Such an importance is measured as the (mass behind the) percentile that the weight of e occupies within the global weight volume at time t. The rationale of using percentiles instead of actual values is to have a fair measure of the relative importance of a weight value with respect to all other weights of the same snapshot. To establish how anomalous the importance of e at time t is, with respect to the past history of e, our model next compares its percentile weight at time t_i with the corresponding percentile at a *reference* past instant t_{i-r}, for a set of reference instants $r \in R$. As an example, if the input horizon $\mathcal{T}$ has a daily granularity, the references $r \in R$ could be weeks/months before. The ultimate anomaly score assigned to e at time t_i is the median difference between the percentile at time t_i and any percentile at time t_{i-r}, $r \in R$.

Whilst being extremely simple, our method has proved to be effective in the experiments. Similar approaches have been widely used in the literature. For example Steiner *et al.* [79] use simple spikes in the concurrent edits to Wikipedia pages to perform detection of breaking news. A similar mechanism has been also been applied as a core ingredient in first-story detection: the traditional approach [2] to this problem detects *first* (i.e., new) stories by comparing news articles to the previous ones, and selecting those news articles whose cosine similarity over tf-idf vectors to its nearest neighbor is less than a threshold. While very simple, this approach effectively outperforms more complex language-model approaches, and it is still a key baseline in recent work about first-story detection, where the focus has been put more on improving efficiency rather than effectiveness [42,58,64], although some

[2] In this work we use the term "story" (or "event") to denote a set of textual units (e.g., words, n-grams, entities, posts) describing a certain real-world event.

latest efforts [89] have also advanced effectiveness using multiple-nearest neighbor clustering.

The pseudocode of our anomaly-detection model is reported as Algorithm 1. The main steps are as follows. We process any time instant in $\mathcal{T}$ one at a time, and, for every $t_i \in \mathcal{T}$, we first scan all edges $e \in E_{t_i}$ and compute its corresponding normalized weight $f'_{t_i}(e)$ as the fraction of its original weight $f_{t_i}(e)$ to the total weight volume $TOT(t_i) = \sum_{e \in E_{t_i}} f_{t_i}(e)$ of time t_i (Lines 1–3). We then sort all edges $e \in E_{t_i}$ according to non-decreasing order of $f'_{t_i}(e)$ (Line 4), and exploit this ordering to compute the percentile $p_{t_i}(e)$ associated with each edge $e \in E_{t_i}$ at time t_i (Line 6). If the reference point t_{i-r} is a valid point in the time horizon $\mathcal{T}$, i.e., $i - r > 0$, then we also compare the percentile $p_{t_i}(e)$ of edge e at time t_i with the corresponding percentile at time t_{i-r}, for all reference instants $r \in R$. If the difference between the two percentiles for a reference instant r is positive, then a score $\phi_{t_i}(e,r)$ equal to such a difference is considered. Otherwise, the score $\phi_{t_i}(e,r)$ is set to zero. Ultimately, edge e is assigned the median of such scores $\{\phi_{t_i}(e,r)\}_{r \in R}$ as an anomalous score at time t_i (Lines 7–15).

As far as running time, the most expensive step of Algorithm 1 is the sorting in Line 4. Hence, denoting by n the number of vertices in the input temporal graph $\mathcal{G} = (V, \{E_t, f_t\}_{t \in \mathcal{T}})$ and by m the maximum number of edges over all snapshots of $\mathcal{G}$, i.e., $n = |V|$ and $m = \max_{t \in \mathcal{T}} |E_t|$, the time complexity of Algorithm 1 is $\mathcal{O}(|\mathcal{T}| m \log n)$.

Algorithm 1 AnomalyScores

Input: A temporal graph $\mathcal{G} = (V, \{E_t, f_t\}_{t \in \mathcal{T}})$, a set $R \subseteq \mathbb{N}^+$ of integers
Output: An anomalous temporal graph $\mathcal{G}^A = (V, \{E_t, \phi_t\}_{t \in \mathcal{T}})$

1: **for all** $t_i \in \mathcal{T}$ **do**
2: $\quad TOT(t_i) \leftarrow \sum_{e \in E_{t_i}} f_{t_i}(e)$
3: $\quad$ for all $e \in E_{t_i}$, let $f'_{t_i}(e) := f_{t_i}(e)/TOT(t_i)$
4: $\quad$ sort edges $e \in E_{t_i}$ by ascending $f'_{t_i}(e)$
5: $\quad$ **for all** $e \in E_{t_i}$ following the order given by $f'_{t_i}(e)$ **do**
6: $\quad\quad p_{t_i}(e) \leftarrow \sum_{e' \in E_{t_i} | f'_{t_i}(e') \leq f'_{t_i}(e)} f'_{t_i}(e')$
7: $\quad\quad$ **for all** $r \in R$ **do**
8: $\quad\quad\quad$ **if** $i - r > 0 \ \wedge \ p_{t_i}(e) > p_{t_{i-r}}(e)$ **then**
9: $\quad\quad\quad\quad \phi_{t_i}(e,r) \leftarrow p_{t_i}(e) - p_{t_{i-r}}(e)$
10: $\quad\quad\quad$ **else**
11: $\quad\quad\quad\quad \phi_{t_i}(e,r) \leftarrow 0$
12: $\quad\quad\quad$ **end if**
13: $\quad\quad$ **end for**
14: $\quad\quad \phi_{t_i}(e) \leftarrow$ median value among $\{\phi_{t_i}(e,r)\}_{r \in R}$
15: $\quad$ **end for**
16: **end for**

2.2. Step 2: Extracting anomalous temporal subgraphs

The second step of our approach to discovering buzzing stories follows the general idea that every piece of data (e.g., a post in a social-networking platform or a query issued to a search engine) related to a specific story typically tends to involve the same set of main objects (e.g., terms or entities). We take the anomalous temporal graph $\mathcal{G}^A$ defined in the previous step, as well as a time window $W \subseteq \mathcal{T}$ that denotes the time period under consideration, and we seek K subgraphs of $\mathcal{G}^A$ that exhibit high density in the window W. To recognize a story as buzzing, it needs to have high cohesiveness among *all objects* therein and for *all time instants* in the window W. Hence, given a subgraph S of $\mathcal{G}^A$ and a time window $W \subseteq \mathcal{T}$, in this work the following definition of cohesiveness it is used:

$$\delta(S, W) \ = \ \min_{u \in S} \ \min_{t \in W} \ deg_S(u, t). \tag{1}$$

Analogously, The overall cohesiveness of a set of subgraphs $\mathcal{S}$ of $\mathcal{G}^A$ is measured by taking the sum of the cohesiveness of each subgraph in $\mathcal{S}$:

$$\Delta(\mathcal{S}, W) \ = \ \sum_{S \in \mathcal{S}} \delta(S, W). \tag{2}$$

The double-min function in Equation (1) allows for capturing the requirements: high cohesiveness among all objects and for all time instants. The minimum over vertices helps mitigate the so-called free-rider effect (vertices attached to a strong group by weak links [16,78]), thus preventing stories from containing undesired outlying objects. At the same time, minimizing over all time instants in W captures the fact that a buzzing story should exhibit high strength of association during the entire period spanned by W. According to [8] a story with too many objects is hard to be processed by a human being. Then, we require that each story/subgraph be limited in size. Each output subgraph S is required to have size no more than an input integer N, with N in the order of a few tens.

Problem statement. Motivated by the above discussion, we now state the problem we aim to solve.

Problem 1. ***(ANOMALOUS TEMPORAL SUBGRAPH DISCOVERY** (ATSD)**)*** *Given an anomalous temporal graph $\mathcal{G}^A = (V, \{E_t, \phi_t\}_{t \in \mathcal{T}})$, a time window $W \subseteq \mathcal{T}$, and two integers $K, N \geq 1$, find a set $\mathcal{S}^* = \{S_1, \ldots S_K\}$ of disjoint subgraphs of $\mathcal{G}^A$ such that (i)*

$\forall i \in [1..K] : |S_i| \leq N$, *and (ii)* $\Delta(\mathcal{S}^*, W)$ *is maximized.* □

Theorem 1. *The* ATSD *problem is* **NP***-hard.*

Proof. We prove **NP**-hardness by reducing from the well-known CLIQUE (decision) problem: given a graph $G = (V, E)$ and an integer k, decide if G contains a clique of size k. We reduce CLIQUE to a special case of ATSD where $|\mathcal{T}| = 1$, $K = 1$, and $\forall t \in \mathcal{T}, e \in E_t : \phi_t(e) = 1$. This special case of ATSD corresponds to having a simple unweighted input graph (i.e., instead of a temporal graph) and asking for one output subgraph. The corresponding decision version is: given a (simple, unweighted) graph $G' = (V, E)$ and two integers N, M, decide if a subgraph with size no more than N and min degree at least M exists in G.

Given an instance $I = \langle G, k \rangle$ of CLIQUE, we construct in polynomial time an instance $I' = \langle G', N, M \rangle$ of (the special version of) ATSD by setting $G' = G$, $N = k$, $M = k-1$. We show that I is a YES-instance for CLIQUE if and only if I' is a YES-instance for ATSD. Indeed, if G contains a clique of size k, this corresponds to a subgraph with $k = N$ vertices and minimum degree $k - 1 = M$. Therefore, this would make the corresponding ATSD instance I' be a YES-instance as well. On the other hand, if G' contains a subgraph of size $N = k$ and minimum degree $M = k - 1$, it means that this subgraph is a clique of size k. □

The DenseTemporal algorithm. As Problem 1 is **NP**-hard, we devise a fast heuristic that yields accurate solutions in practice, as confirmed by our experiments in Section 3. To properly design the proposed heuristic, we first introduce a simplified version of the ATSD problem, termed UNBOUNDED-ATSD (U-ATSD), where only one subgraph is required as output ($K = 1$) and the size of the output subgraph is left unbounded ($N = \infty$). We show that a polynomial-time (exact) algorithm exists for the U-ATSD problem, and use such an algorithm as a basis for the proposed heuristic.

Problem 2. *(*UNBOUNDED ANOMALOUS TEMPORAL SUBGRAPH DISCOVERY *(*U-ATSD *)) Given an anomalous temporal graph* $\mathcal{G}^A = (V, \{E_t, \phi_t\}_{t \in \mathcal{T}})$ *and a time window* $W \subseteq \mathcal{T}$*, find a subgraph* S^* *of* $\mathcal{G}^A$ *that maximizes* $\delta(S^*, W)$. □

This simplified version of the ATSD problem resembles the problem of finding the *inner-most core* in a graph (and the notion of *core decomposition*) [75], which we briefly recall below.

Algorithm 2 U-ATS

Input: An anomalous temporal graph $\mathcal{G}^A = (V, \{E_t, \phi_t\}_{t \in \mathcal{T}})$, a time window W.
Output: A subset of vertices (subgraph) $S^* \subseteq V$.

$\mathbf{c} \leftarrow \emptyset$, $\mathbf{Q} \leftarrow \emptyset$
for all $u \in V$ **do**
  $p(u) \leftarrow \min_{t \in W} deg(u, t)$
  insert u in $\mathbf{Q}$ with priority score $p(u)$
end for
$k \leftarrow 0$
while $\mathbf{Q} \neq \emptyset$ **do**
  $u \leftarrow$ highest-priority vertex in $\mathbf{Q}$
  $p(u) \leftarrow$ priority score of u in $\mathbf{Q}$
  if $p(u) > k$ **then**
    $k \leftarrow p(u)$
  end if
  $\mathbf{c}[u] \leftarrow k$
{update priority queue}
  for all $t \in W, v \in \mathbf{Q} \mid (u, v) \in E_t$ **do**
    $deg(v, t) \leftarrow deg(v, t) - \phi_t(u, v)$
  end for
  for all $t \in W, v \in \mathbf{Q} \mid (u, v) \in E_t$ **do**
    $p(v) \leftarrow$ priority score of v in $\mathbf{Q}$
    $p'(v) \leftarrow \min_{t \in W} deg(v, t)$
    update order of v in $\mathbf{Q}$ based on the new priority score $p'(v)$ (if $p'(v) \neq p(v)$)
  end for
  remove u from $\mathcal{G}^A$
end while
$S^* \leftarrow \{u \in V \mid \mathbf{c}[u] = k\}$

The *k-core* (or *core* of order k) of a graph G is defined as the maximal subgraph in which every vertex is connected to at least k other vertices within that subgraph. The set of all cores, for all $k \in [1..k^*]$, forms the *core decomposition* of G. The linear time algorithm proposed by Batagelj and Zaveršnik [13] iteratively removes the smallest-degree vertex from the graph and sets the core number of the removed vertex accordingly.

The U-ATSD problem resembles the problem of extracting the inner-most core of a graph, but it comes with two additional challenges: (*i*) our input is a temporal graph composed of multiple snapshots, and (*ii*) the maximization of the min degree should be ensured for all snapshots corresponding to the instants in the given time window. Despite being more complicated than inner-most-core extraction, the U-ATSD problem can still be solved in polynomial time.

The algorithm to solve the U-ATSD problem is inspired by the one by Batagelj and Zaveršnik, where the vertex to be removed at each step is the one with min-

Algorithm 3 DenseTemporal

Input: An anomalous temporal graph $\mathcal{G}^A = (V, \{E_t, \phi_t\}_{t\in\mathcal{T}})$, a time window W, two integers $K, N \geq 1$.

Output: A set $\mathcal{S}^* = \{S_i\}_{i=1}^K$ of K disjoint subgraphs of $\mathcal{G}^A$, with $|S| \leq N$, $\forall S \in \mathcal{S}^*$.

$\mathcal{S}^* \leftarrow \emptyset$
while $|\mathcal{S}^*| < K$ **do**
  $S \leftarrow$ U-ATS $(\mathcal{G}^A, W)$ {Algorithm 2}
  if $|S| > N$ **then**
    run the min-degree-vertex removal phase of Algorithm 2 on S until it becomes empty and generate a set of subgraphs $\mathcal{S} = \{\hat{S}_1, \ldots, \hat{S}_{|S|}\}$, with $\hat{S}_1 = S$
    $S \leftarrow \text{argmax}_{i\in[|S|-N+1..|S|]}\, \delta(\hat{S}_i, W)$
  end if
  remove the subgraph induced by S from $\mathcal{G}^A$
  $\mathcal{S}^* \leftarrow \mathcal{S}^* \cup \{S\}$
end while

imum weighted degree in the whole time window W, i.e., a vertex u minimizing $\min_{t\in W} deg_{\mathcal{G}'}(u,t)$, where $\mathcal{G}'$ is the anomalous temporal graph at the current iteration. Once a vertex has been removed, the degree of all its neighbors in all the snapshots of the time window needs to be updated and the new vertex with minimum degree needs to be identified. To do this efficiently, one can employ a priority queue $\mathbf{Q}$ where the basic operations of insertion, deletion, and update are performed in time logarithmic in the size of the queue. The pseudocode of the U-ATS algorithm is reported as Algorithm 2.

The time complexity of Algorithm 2 is $\mathcal{O}(|W|m\log n)$ ($n = |V|, m = \max_{t\in\mathcal{T}} |E_t|$). Indeed, each vertex in the graph and its corresponding neighbors in each snapshot are visited only once. This means that, the overall number of operations after all vertices have been processed is $\mathcal{O}(|W|m)$. This cost should be multiplied by a logarithmic factor due to the maintenance of the priority queue.

Finally, in the next theorem we formally show the soundness of the algorithm.

Theorem 2. *Algorithm 2 returns a solution to Problem 2.*

Proof. A vertex property function on a graph $G = (V, E)$ is a function $g : V \times 2^V \to \mathbb{R}$. A vertex property function g is said *monotone* if for all $C_1, C_2 \subseteq V : C_1 \subseteq C_2$ it holds that $\forall v \in V : g(v, C_1) \leq g(v, C_2)$ [13]. Let $\mathcal{G}^A = (V, \{E_t, \phi_t\}_{t\in\mathcal{T}})$ be an anomalous temporal graph and let W be a time window. For any vertex $u \in V$ and subgraph $S \subseteq V$, let g be defined as $g(u, S) := \min_{t\in W} deg_S(u,t)$. The vertex property function g defined this way corresponds to the property at the basis of the inner-most core to be output by the U-ATSD problem (Equation (1)). It is easy to see that this property function is monotone, as the weights on the edges of $\mathcal{G}^A$ are non-negative, hence the min degree (over all instants in W) in a subgraph S is no less than the corresponding min degree in a supergraph of S. The proof is completed by the Batagelj and Zaveršnik result [13]: for a monotone vertex property function g, the algorithm that repeatedly removes a vertex with the smallest g value correctly determines cores based on g. □

Algorithm 4 Buzz

Input: A temporal graph $\mathcal{G} = (V, \{E_t, f_t\}_{t\in\mathcal{T}})$, a time window W, a set $R \subseteq \mathbb{N}^+$ of integers, two integers $K, N \geq 1$.

Output: A set $\mathcal{S}^*$ of K subsets of vertices of $\mathcal{G}$.

generate an anomalous temporal graph $\mathcal{G}^A$ by running Algorithm 1 on input $\langle \mathcal{G}, R\rangle$
get $\mathcal{S}^*$ by running Algorithm 3 on input $\langle \mathcal{G}^A, W, K, N\rangle$

The U-ATS algorithm provides a solid basis for solving the general ATSD problem. The method we propose is indeed an extension of U-ATS where we ask for two additional requirements: (*i*) the output subgraph(s) should be bounded in size, and (*ii*) multiple subgraphs need to be output. The first requirement is met by keeping iterating the min-degree-vertex removal phase of the U-ATS algorithm until we are left with an empty graph. This procedure generates a set of subgraphs. The subgraph with highest density δ among the ones with size at most N is output. As far as outputting multiple subgraphs, we adopt an intuitive strategy where, once the first subgraph has been found, it is removed from the graph, and the next subgraph is identified by running the algorithm on the remaining graph, until K subgraphs have been extracted. All steps of the proposed algorithm are in Algorithm 3. The time complexity of the algorithm is K times the time complexity of U-ATS, that is $\mathcal{O}(K|W|m\log n)$.

2.3. The overall Buzz approach

The overall approach we propose to identify buzzing stories is summarized in Algorithm 4. The algorithm consists in sequentially running the aforementioned Step 1 and Step 2, and its overall time complexity is $\mathcal{O}((|\mathcal{T}| + K|W|) \times m\log n)$, with n being the number of vertices in the input graph and $m = \max_{t\in\mathcal{T}} |E_t|$.

Step 1 can be performed offline and then be updated incrementally for every new time instant. At query time, we only need to perform Step 2, which leads to an online time complexity of only $\mathcal{O}((K|W|) \times m \log n)$.

3. Evaluation on search-log data

In this section we describe the empirical evaluation we conducted on a dataset extracted from a real-world web-search log.

Dataset. We employed a query log of a popular commercial web-search engine.[3] Web-search queries have traditionally been used in the story-identification literature [55,98]. Indeed, relevant real-world events raise interest/concern in people, who naturally turn to search engines to gather information. This renders online searches a valuable source to seek buzzing stories. We analyzed an anonymized sample of that query log, spanning about 18 months from 2013-2014. From this dataset, which we dub Q_{Log}, we derived a temporal graph $\mathcal{G}$ and an anomalous temporal graph $\mathcal{G}^A$. We point out that the proposed method is general enough to be applied to any other type of user-generated content, such as data from microblogs/social networks. We defer the use of other datasets to future work.

Building the $\mathcal{G}$ graph. The Q_{Log} dataset spans a time horizon $\mathcal{T}$ of 558 days, and contains hundreds of billions of queries, with tens of millions of distinct terms. To filter out noise, we pre-processed Q_{Log} retaining, for every day $t \in \mathcal{T}$, only the queries with at least 50 occurrences. We derived from Q_{Log} a temporal graph $\mathcal{G} = (V, \{E_t, f_t\}_{t\in\mathcal{T}})$, consisting of daily snapshots. The snapshot (E_t, f_t) of each day $t \in \mathcal{T}$ was extracted from the set Q_t of all queries submitted at day t, with the respective number of occurrences. From each query $q \in Q_t$ we extracted all distinct pairs of non-stop-word terms, and built the edge set E_t as the set of term pairs co-occurring in at least one query $q \in Q_t$. Each edge (u, v) was assigned a raw weight $f_t(u, v)$ equal to the sum of the occurrences of all originating queries from Q_t where u and v are both present.

Building the anomalous $\mathcal{G}^A$ graph. We built the anomalous temporal graph $\mathcal{G}^A$ from the *raw* temporal graph $\mathcal{G}$ by running the algorithm AnomalyScores (see Section 2.1) with $R = \{7\}$, i.e., using a single reference time instant set to one week before. The choice of R (as well as the length of the time window) obviously impact the type of events that we detect. Local and small-scaled events might require smaller slots and finer granularity. However, in line with related work [80,81], we are interested in world-wide stories with a lasting impact on social-media users. All graphs were built with a Hadoop implementation of the above process, exploiting a cluster of 500 nodes. Table 1 reports statistics on $\mathcal{G}^A$ and $\mathcal{G}$.

[3] Yahoo Web Search

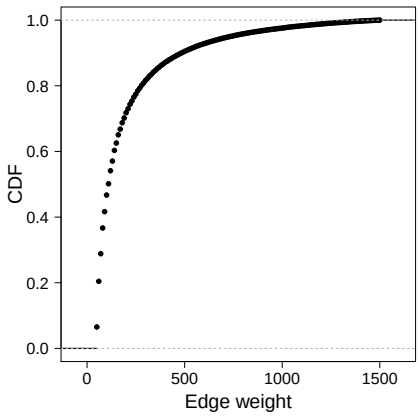

Fig. 1. Search-log evaluation: CDF for edge weights of $\mathcal{G}$.

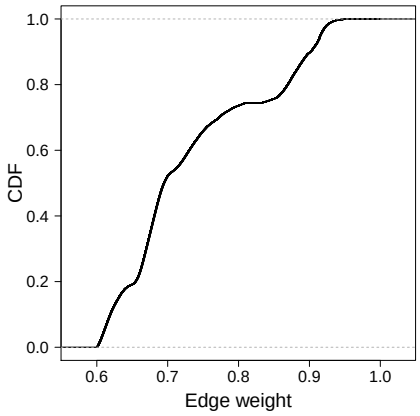

Fig. 2. Search-log evaluation: CDF for edge weights of $\mathcal{G}^A$.

Graph characterization. We highlight here some interesting characteristics of the temporal graph $\mathcal{G}$ and the anomalous temporal graph $\mathcal{G}^A$. Figures 1–2 show the cumulative distribution (CDF) of edge weights of graphs $\mathcal{G}$ and $\mathcal{G}^A$, respectively. For graph $\mathcal{G}$, we observe that most edge weights falls between 50 (the minimum number of daily co-occurrences, due to the chosen η threshold), and 1 500. The average maximum score across all snapshots is 3 803 170, whereas the average

Table 1
Search-log evaluation: statistics of the temporal graphs.

	Vertices		**Edges**	
	$\mathcal{G}$	$\mathcal{G}^A$	$\mathcal{G}$	$\mathcal{G}^A$
Mean	6 933 237	1 554 728	129 771 466	9 849 487
SD	0	628 347	0	5 402 513

mean is 399.85. In the case of $\mathcal{G}^A$, edge weights appear to be more evenly distributed in the interval $[0, 1]$.

A second difference between $\mathcal{G}$ and $\mathcal{G}^A$ regards the size of the snapshots. For both graphs Table 1 reports the mean value and standard deviation for the number of vertices and edges across all snapshots. Note that the number of vertices and edges do not vary in the snapshots of $\mathcal{G}$. Replacing the raw weights with anomaly scores (to derive $\mathcal{G}^A$ from $\mathcal{G}$) causes remarkable reductions in size: 22% of the vertices and 7% of the edges of $\mathcal{G}$ are retained. This is a nice side effect due to the adoption of an anomaly model, which automatically filters our those associations that are not worth considering for the task of detecting buzzing stories.

Competitors. We compared our Buzz method to the two main approaches discussed in the Introduction. The first approach builds a graph modeling the association between domain objects and looks for cohesive subgraphs in it, without considering deviations (anomalies) from the normal level observed over the entire time horizon [12,26,74,92,98]. In our context this corresponds to running Algorithm 3 on the original graph $\mathcal{G}$, and using a time-window size $|W| = 1$, whose unique instant corresponds to the day where stories are identified. We refer to this method as RGB (raw-graph baseline).

The second approach applies an anomaly model to characterize abnormal associations between domain objects. It ignores object associations (i.e., it exploits no co-association graph), and identifies stories by a-posteriori grouping objects with similar anomalous behavior. Specifically, as a representative of this category, we considered SAX* [80].

Testbed. We considered the temporal graph $\mathcal{G}$ and the anomalous temporal graph $\mathcal{G}^A$ extracted from Q_{Log}, as described above. We evaluated the proposed Buzz and the SAX* and RGB competitors on a test set of 50 days, which were sampled uniformly at random from the whole horizon $\mathcal{T}$ of 558 days spanned by $\mathcal{G}$ and $\mathcal{G}^A$. For each selected date, we ran Buzz on $\mathcal{G}^A$, RGB on $\mathcal{G}$, and SAX* on the corresponding time series of occurrences of individual terms.We varied window size W (starting in the given date), maximum size N of each output subgraph, and maximum number K of output subgraphs as follows:

- $|W| \in \{1, 2, 3, 4, 5\}$;
- $N \in \{10, 15, 20, 25, 30\}$;
- $K \in \{10, 15, 20, 25, 30\}$.

Testing 5 values for each parameter led to a total of 125 different configurations to be given as input to Buzz and RGB. In the case of SAX*, instead, the only parameter that is defined is the window size $|W|$. Indeed, this algorithm allows for specifying neither the number N of stories nor the story size K. To ensure a fair comparison between Buzz and RGB vs. SAX*, for a given value of N and K, we thus retained the SAX* stories with size no more than N, and, if SAX* had output more than K stories, we sampled a random subset of size K. For the sake of robustness, the sampling procedure was repeated 10 times and performance indicators were obtained by averaging across the 10 samples.

3.1. Anecdotal evidence

In Tables 2–4 we show some examples of buzzing stories extracted by the proposed Buzz, and the RGB and SAX* baselines, respectively. Table 2 shows that Buzz tends to extract real-world events on different topics — sport, politics, or show business — that became buzzing in those test days. A number of stories are about sport: Cristiano Ronaldo winner of the 2014 Baloon d'Or (Example #1); the open ceremony of Sochi 2014 Olympic Winter Games (Example #3); the gold medal of Yulia Lipnitskaya, a fifteen-year old Russian prodigy in figure skating (Example #4); the perfect 10.0 scored by the gymnast Lloimincia Hall for her routine against Alabama, whose performance went viral (> 850K views online) in April 2014 (Example #11). Another bunch of stories (Examples #5–#8) deal with natural disasters or catastrophic events: the protests in Ukraine, the Costa Concordia cruise disaster, two tornadoes that bore down two towns in northeast Nebraska, and the disappearance of Malaysia Airlines Flight 370. Example #9 is focused on the primary victory of congressman Charlie Rangel of New York, after facing one of the most serious challenges of his career, while Example #10 concerns a presumed sighting of a deer-like UFO over the city of Kowloon in China. Varying the window size $|W|$ seems to impact the type of event detected. For instance, Example #12 testifies that a larger $|W|$ (5 in this case) allows for capturing particular aspects of very popular events, like

Table 2

Search-log evaluation: examples of stories detected by the proposed Buzz method.

#	Date	$\|W\|$	N	Story
1	2014-01-13	1	10	cristiano dor wins ronaldo fifa ballon
2	2014-01-28	2	25	mexico templar treasure knights
3	2014-02-07	3	10	sochi russian nbc opening watch ceremony
4	2014-02-09	3	10	day figure russia julia skating medal ceremony
5	2014-02-19	2	25	protests live ukrainian police kiev
6	2014-02-27	2	30	captains costa wreck concordia
7	2014-06-16	3	25	nebraska failure llc tornado monday big
8	2014-03-12	3	15	crash malaysian plane flight mh370 missing
9	2014-06-25	1	25	charlie rangel primary election
10	2014-04-06	2	20	ufo deer nasa people kowloon sightings china
11	2014-04-10	2	25	gymnast lloimincia legs hall girls alabama
12	2014-03-03	5	10	acceptance jared speech leto novak oscars goldie
13	2014-01-13	1	30	gracie ashley progeria parents scott berns
14	2014-01-13	2	30	scott progeria death berns
15	2014-01-13	3	10	search papa baby progeria death berns
16	2014-01-13	4	10	pictures progeria death berns

Table 3

Search-log evaluation: examples of stories detected by the RGB competing method.

#	Date	N	Story
1	2014-01-13	10	earthquake rico puerto
2	2014-01-28	20	grammys 2014 monica lewinsky
3	2014-02-07	30	sochi ceremony opening olympics
4	2014-02-09	30	count medal sochi olympics skating figure young girl
5	2014-02-19	20	lansbury angela
6	2014-02-27	20	costa concordia
7	2014-06-16	30	happy fathers day pictures funny lebron james
8	2014-03-12	30	mh370 flight malaysia airlines
9	2014-06-25	10	bieber justin selena gomez grande ariana
10	2014-04-06	20	ufo sightings
11	2014-04-10	20	lsu gymnast
12	2014-03-03	30	leto jared

Jared Leto's impressive acceptance speech at the 2014 Oscars ceremony. Similarly, Examples #13–#16 show a natural tendency of our Buzz method to capture different aspects of the same key event. All of these four examples are about the death of teenager Sam Bern, which was caused by progeria disease, but varying the size of the time-window leads to different additional terms corresponding to different facets of the story. Tables 3 and 4 show that RGB and SAX* are to some extent able to detect events that are similar in spirit to the ones detected by our Buzz. However, both competitors exhibit a critical weakness: RGB has a tendency to extract ever-popular topics, such as full names of celebrities or searches for funny pictures, while SAX* often combines (erroneously) multiple events in a sin-

Table 4
Search-log evaluation: examples of stories detected by the SAX* competing method.

#	Date	Story
1	2014-01-13	community diet equipment helen
2	2014-01-28	lynch rosie started cadillac created torres trading uss automated beckinsale blanchett bodies cate coronado faris forex greta hawaii kaling kate katrina knights miller pete required review reviews robot robots seeger sienna software templar
3	2014-02-07	delivery divorce seymour thompson wife buy forum
4	2014-02-09	bras easter engagement jean laser petite posters rod davis death dia earn ellen evelyn gifts jackie linda making meryl michael money nike palm prison robinson skater skaters skating slips speed tanya thrones tools types valentines walking 1990 anderson beatles bmw charlie colored concert crawford
5	2014-02-19	verde component configures detail fuck god quotations expedition gravity johnny mao michelle minibb mvnforum plymouth scout seuss ukraine ukrainian vbulletin app artwork blackberry brazzers civic classroom
6	2014-02-27	buffalo gordon jacket katy perry sale stevens survivor tebow travis warship wilson alyssa ammo ammunition barrymore blog blogs bulk bullock cheap concordia costa drew fmj hudson journal leah mara mask oscar plane remini russian sandra singles
7	2014-06-16	pamela playing tornado johnny original
8	2014-03-12	young holiday university sites cookies crime flight mh370 rob scene
9	2014-06-25	stock store dicaprio fanny leonardo aaron collins
10	2014-04-06	jessica station watch chocolate east
11	2014-04-10	victorian obama single
12	2014-03-03	internet riley search stars adobe beth lara nudity brad brazil carpet cate channing concept degeneres dressed ellen farmiga garner goldie gomez hawn jared jennette job johansson kardashian kendrick kim kinney lawrence leto liza loss lupita margot matthew mccurdy minnelli museum norman novak nyong olivia oscar oscars pitt portia robbie roberts rossi

gle one, likely due to the fact that SAX* does not admit any bound on the size of the output stories.

3.2. Evaluation: Anomalous nature of the stories

Buzzing stories should possess two main characteristics: (*i*) they should be *anomalous* enough, (*ii*) they should match real events that took place in the time window considered. The goal of our evaluation is to assess how good each set of terms (subgraph) S output by any considered method is with respect to these two different aspects. In the following we focus on the first aspect, while the second aspect will be discussed in the next subsection. Particularly, for the first aspect we checked that the story does not match a concept that is regularly searched by the web crowd (*rarity* of the event). To this end, we involved two metrics: (*i*) *search frequency* in Q_{Log}, (*ii*) *inter-day similarity*.

Search frequency. For each output story we checked how much and how regularly it was searched within Q_{Log} in the time horizon $\mathcal{T}$. The rationale is that an anomalous story should not be too frequent. To soundly seek matches in Q_{Log}, we processed queries and stories by removing stop words and non-alphanumeric characters, performing stemming [70], and sorting the stemmed terms lexicographically. For each story, we computed the number of distinct days it occurs in at least one query of that day, and the average frequency over its daily occurrences. For each method, we then computed avg and max of these counts over all buzzing stories and reported them in Table 5. A striking difference exists among RGB on one side, and SAX* and Buzz on the other side. RGB finds stories correspond-

Table 5
Search-log evaluation: search frequency of the buzzing stories.

Method	Measure	Mean	Max.
RGB	**NumDays**	390.7	558
	Mean Freq	56 550	368 000
SAX*	**NumDays**	0.169	383
	Mean Freq	1.184	756.5
Buzz	**NumDays**	3.609	558
	Mean Freq	213.5	91 040

ing to over-popular searches: half of the RGB's stories appear in the log almost every day (552 days over a total of 558), and with high frequency (max story frequencies above $13M$). Indeed, by manual inspection we verified that many of these correspond to celebrities or navigational queries. SAX* and Buzz are comparable to each other and behave very differently from RGB: they extract sets of terms that occur seldom, i.e., the average number of distinct days they appear in the log is 0.2 and 3.6, respectively. In conclusion, employing an anomaly-detection model, which is a common trait for SAX* and Buzz but not for RGB, appears to be critical to avoid the pitfall of retrieving over-popular topics, and instead identify buzzing stories.

Ever-present searches and popular stories capture people's ordinary habits, preferences, and everyday activities. They fall within a different area of interest, which Pink *et al.* [68] have recently put under the concept of *mundane data*. Mundane data is that emerging from the ordinary, usually un-noticed and below the surface routines, contingencies, and accomplishments of our everyday life. Pink *et al.* argue that studying mundane data is important for a plethora of reasons, including the fact that the mundane is a domain of creativity and improvisation, and an inseparable and undeniable part of our life; as such, it represents a critical source of information to advance social sciences and to assist designers and policy markers who create digital interventions in everyday life contexts, like energy demand reduction [67] or promoting health [82] through the presentation of data about their everyday practices and bodies to consumers.

By reflecting ordinary and routinary aspects of life, popular stories capture a different angle of the digital world, than the one that discovering buzzing stories investigates. Being explicit designed to identify *unexpected* and *anomalous* events, our work rather connects with those research efforts that focus on *big crisis data* [21], i.e., they deal with what is novel, spectacular, disruptive or revolutionary. Our algorithm could be a useful contribution to the efforts aimed to exploit social-media data for improving the handling of emergencies and crisis situations, which may unfortunately arise in a variety of domains (e.g., natural disasters, city malfunctioning events, terror attacks).

Whilst the popular, ordinary, mundane must not necessarily be conceptualized as opposed to the spectacular or extraordinary, it is clearly a different angle/dimension, although with possible relations.

Inter-day similarity. As a second metric, we examined how each method tends to extract the same stories for different dates. The desideratum is that this does not happen, as an anomalous story should not be too frequent. We tested this by considering all possible pairs of (not necessarily consecutive) days in our test set of 50 dates, and, for each pair of days, we computed the Jaccard similarity (counting the bag of words of each story as a distinct item) between the sets of stories of each parameter configuration. Results (averaged over all comparisons for a configuration and over all configurations) are presented in Table 6. Once again, RGB behaves very differently from the two other methods. For Buzz and SAX* the average Jaccard similarity is always (almost) zero: this is consistent with the fact that anomalous stories should not appear repeatedly over time. On the other hand, similarity among RGB's stories is much higher, which further testifies its *non-anomalous* nature.

Table 6

Search-log evaluation: average Jaccard coefficient between sets of stories extracted in different days.

$\|W\|$	RGB	SAX*	Buzz
1	0.0468	0.0000	0.0001
2	0.1145	0.0000	0.0000
3	0.1808	0.0000	0.0000
4	0.2141	0.0000	0.0000
5	0.2306	0.0000	0.0000

Table 7

Search-log evaluation: editorial assessment.

Method	# Events	YES Events		NO Events	
		#	%	#	%
ALL	464	272	58.6	192	41.4
SAX*	144	60	44.4	80	55.6
RGB	160	87	54.5	73	45.6
Buzz	160	**121**	**75.6**	39	24.2

3.3. Evaluation: Correspondence with real-world events

The second part of our evaluation was devoted to assessing whether the detected stories match real-world events, which we did by conducting (i) an editorial study with human assessors, and (ii) an automated quantitative evaluation.

Editorial assessment. We recruited three human judges and asked them to provide a YES/NO answer to the

Table 8
Search-log evaluation: correspondence with real-world events.

Parameter		RGB avg cosine	SAX* avg cosine	Buzz avg cosine	% Variation Buzz vs. SAX*
$\|W\|$	1	0.343	0.101	0.062	−39.1 %
	2	0.370	0.107	0.128	19.8 %
	3	0.305	0.071	0.109	53.3 %
	4	0.281	0.030	0.077	156.6 %
	5	0.199	0.010	0.058	475.7 %
N	10	0.299	0.064	0.120	86.9 %
	15	0.297	0.064	0.092	43.7 %
	20	0.300	0.064	0.077	20.9 %
	25	0.301	0.064	0.074	15.8 %
	30	0.301	0.064	0.071	11.8 %
K	10	0.303	0.075	0.101	34.8 %
	15	0.301	0.068	0.094	37.1 %
	20	0.300	0.063	0.087	36.9 %
	25	0.298	0.059	0.080	36.3 %
	30	0.297	0.055	0.073	34.1 %

question: "Does the story match a real event?" We encouraged editors to query their preferred search engine with the terms and dates of a story, and explore the corresponding results. Given that the labeling was complex and time consuming, the assessment was conducted on a sample of our test set. Specifically, we randomly picked 16 < Date, $|W|$, N > configurations and fixed $K = 10$. This led to a total of 464 candidate stories, 160 of which were extracted by Buzz, 160 by RGB, and 144 by SAX*. SAX* returned less stories as it does not allow for specifying the number of output stories, and it found less than 10 events for some configurations. For each candidate event, editors were shown the words of the story and the dates in the time window. The stories returned by different methods were randomly mixed. Each judge was asked to assess all 464 candidate events in our sample. Hence, the editorial evaluation provided us with 3 labels for each story. Each story was assigned the label that was chosen by at least two editors.

Table 7 summarizes the results, which show that our Buzz evidently outperforms its competitors. We measured the agreement among editors with the well-established Fleiss' Kappa measure. Our task was quite complex and subjective, thus we expected the inter-annotator agreement to be relatively low. Nevertheless, we obtained a Fleiss' Kappa value of 0.254, which is customarily interpreted as a "fair" level of agreement and thus demonstrates the appropriateness of the study.

Quantitative evaluation. We adopted an automated version of the methodology in [80]: for any buzzing story, we issued a web-search query composed of the terms of the story, we retrieved the top result pages, and evaluated their quality in terms of relatedness to the event. Again, the intuition is that if one queries a search engine on a real event, the top results should be recognized as related to the issued query.

For each detected story we formulated a query with the terms of the story, plus all dates in $[t-1, t+|W|-1]$, where t is the input date and W is the specified time window. For each query, we collected the top ten result pages from the public API of a popular commercial search engine. We represented each result as a bag of words, aggregating title, snippet, and the last part of the url corresponding to the page name. For each buzzing story we computed the cosine similarity between its TF/IDF vector and the TF/IDF vector of each result page, and averaged over the ten results. This way, a higher cosine similarity is an indicator of higher pertinence of the web-search results to the detected story, and, as such, higher correspondence to a real event.

Performance comparison. Table 8 shows the outcome of this experiment. Results for a parameter value

were obtained by averaging over all other parameters. The highest similarity is achieved by RGB. However, based on the evaluation of the anomalous nature of the stories, it is apparent that this mainly depends on the inability of RGB in quantifying the anomaly of a story, and not on a real superiority in detecting buzzing stories. Indeed, RGB mostly extracts term sets matching very popular searches, such as gossip around celebrities, which constantly raise attention over time, and thus cannot be considered as buzzing. Also, these popular queries are typically short (e.g., just celebrity name), hence it is much easier to find search results matching all terms in the story and achieve a higher similarity. As a result, the only meaningful comparison for this assessment is the one between Buzz and SAX*.

Table 8 shows that Buzz clearly outperforms SAX*. The only case in which we observe a loss is for window size $|W| = 1$, which basically means asking for a story that is anomalous during one day only. This is not a serious issue but rather a limit case in our setting, where we target stories raising an anomalous interest over a generally longer period. For $|W| > 1$, Buzz always wins over SAX*. The average gain of Buzz decreases as the maximum story size N increases. This is expected: if a story has more terms, it is less likely that a good match with a snippet is found. Conversely, the gain increases with the number K of stories. This is likely due to the fact that SAX* is often unable to retrieve the number of stories requested. The average running times of the online processing are 1.3 s for Buzz, 1.5 s for SAX*, and 5.9 s for RGB.

4. Evaluation on news data

This section present the evaluation that we conducted on a large corpus of news data. We start by detailing all the phases of the dataset-construction process (Sections 4.1): news collection, preprocessing, news annotation/information-extraction, temporal-graph building. Then, we report some statistics on the constructed dataset (Section 4.2). Finally, we present the testbed built to evaluate our Buzz algorithm and its competitors, as well as the results of the evaluation (Sections 4.3-4.4).

4.1. Dataset construction

News collection and preprocessing. We collected news from the RSS feeds of a list of major Italian online newspapers, which we report in Table 9. We considered a time horizon spanning roughly three months, precisely from December 12th, 2016, to March 7th, 2017. News were collected by exploiting the news-crawling, RSS-feed-processing, and data-cleaning functionalities embedded in the Hermes tool [19]. In particular, as a main data-cleaning operation, we exploited the capability of Hermes to extract the pure textual content of the news, by identifying and removing non-textual content and/or irrelevant content, such as metadata or markups. All the news resulting from the pre-processing phase constitute the set of news we ultimately used as input for the construction of our collection. We hereinafter denote such a set by $\mathcal{D}$. Figure 3 depicts the number of news collected in the various dates of the considered period. The overall number of news is 88 092.

News annotation/information extraction. The next step of our dataset-construction process consisted in extracting from each news in $\mathcal{D}$ useful information that can be exploited to build the ultimate temporal graphs.

In particular, we resorted to an *entity-based representation* of news items, which was derived by extracting *entities* from news. This corresponds to solving a classic NLP task, termed *Entity Recognition and Disambiguation* (ERD), whose goal is to identify entity mentions in a text (entity recognition) and link them to a proper entity of a given knowledge base (entity disambiguation) [77]. To build the collection, we solved the ERD task by resorting to the well-known *wikification* approach, which was first proposed by Mihalcea *et al.* [60], and then has had a huge success in the NLP community [28,36,45]. The wikification ERD method employs Wikipedia as a knowledge base: each article in Wikipedia is considered as an entity, and the anchor text of all hyperlinks pointing to that article constitute the possible mentions for that entity. All entities are organized in a (directed) graph structure given by the underlying Wikipedia hyperlink graph, where vertices correspond to entities and an arc from entity e_1 to entity e_2 exists if e_1 contains an hyperlink to e_2 in its body. In the wikification process the entity-recognition subtask is easily performed by generating all n-grams occurring in the input text and looking them up in a table that maps Wikipedia anchor-texts to their possible candidate entities.[4] For the entity-disambiguation subtask we employ the popular voting approach of Ferragina *et al.* [28], dubbed *Tagme*. For each news in $\mathcal{D}$ we

[4] We generate up to 5-grams.

Table 9
News evaluation: newspapers used to build the dataset.

viaggiaresicuri.it	ilsole24ore.com	it.reuters.com
ingv.it	tg24.sky.it	interno.gov.it
agi.it	ladige.it	ansa.it
ilmessaggero.it	corriere.it	lastampa.it
esteri.it	milanofinanza.it	gazzettadiparma.it
protezionecivile.gov.it	ilfattoquotidiano.it	rai.it
ilgiornale.it	repubblica.it	ilmattino.it
tgcom24.mediaset.it	lagazzettadelmezzogiorno.it	

define its entity-based representation as the set of all its extracted Wikipedia entities that do not match any stop word. Moreover, based on a careful analysis of the distribution of the frequency of an entity within the news collection $\mathcal{D}$, we also discard all entities whose frequency is larger than 3 600. The set of all entities belonging to the entity-representation of a news in $\mathcal{D}$ forms the *entity vocabulary* $\mathcal{V}_e$.

Building the temporal graphs. The entity-based representations of the news in $\mathcal{D}$, along with the entity vocabulary $\mathcal{V}_e$, were exploited to derive a temporal graph with daily granularity. Each news in $\mathcal{D}$ was assigned a timestamp corresponding to the time it has been published. We considered the whole time period spanned by the timestamps of all news in $\mathcal{D}$, and defined our time horizon $\mathcal{T}$ by splitting such a period in fixed intervals of 1 day.

The temporal graph $\mathcal{G}$ was defined as follows. The vertex set of $\mathcal{G}$ corresponds to the entity vocabulary $\mathcal{V}_e$. For each time instant (day) $t \in \mathcal{T}$, the corresponding edge set E_t is defined based on all co-occurences of any two entities in a news whose timestamp belongs to the interval $[t_i, t_{i+1})$. Formally, for any two entities $u, v \in \mathcal{V}_e$, we count all news whose timestamp lies within $[t_i, t_{i+1})$ where u and v co-occur. Let $c_t(u,v)$ denote such a count. We draw an edge (u,v) (and add it to E_t) between all pairs of entities u, v such that $c_t(u,v) \geq \eta$, where η is a threshold defining when the association between two entities is recognized as strong enough.[5] The weight of an edge (u,v) is set as $w_t(u,v) = c_t(u,v)$.

The anomalous temporal graph $\mathcal{G}^A$ was built from the *raw* temporal graph $\mathcal{G}$ by running the algorithm AnomalyScores (see Section 2.1) with $R = \{7, 10, 14\}$,

[5] In our evaluation we set $\eta = 2$.

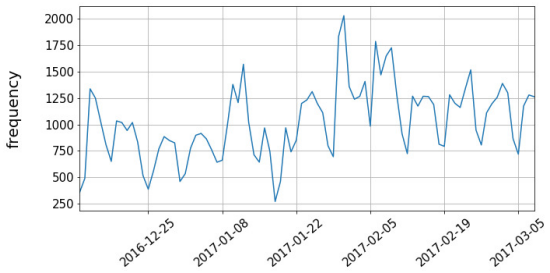

Fig. 3. News evaluation: number of news for each date of the selected time period.

Table 10
News evaluation: statistics of the temporal graphs.

#time instants	86
avg #news	1 024
#non-singleton vertices	1 822
#edges	16 570
min degree	1
avg degree	15.57
median degree	11.27
max degree	193.24

i.e., using three reference time instants set to one week, ten days, and two weeks before.

4.2. Dataset characterization

Table 10 reports aggregated statistics on the temporal graphs we generated: number of time instants, average number of news per time instant, number of (non-singleton) vertices, number of edges, and minimum/average/median/maximum degree of a vertex.

Also, Figure 4 shows the distribution of number of (non-singleton) vertices and number of edges across all the instants in the temporal graph.

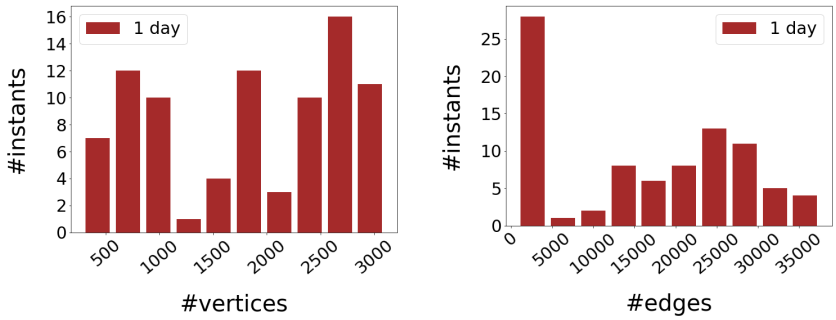

Fig. 4. News evaluation: distribution of number of vertices and edges of the temporal graph.

4.3. Testbed

We considered the temporal graph $\mathcal{G}$ and the anomalous temporal graph $\mathcal{G}^A$ extracted from the news corpus, as described above. We evaluated the proposed Buzz and the SAX* and RGB competitors on a test set of 24 days, which were sampled uniformly at random from the whole horizon $\mathcal{T}$ of 86 days spanned by $\mathcal{G}$ and $\mathcal{G}^A$. For each selected date, we ran Buzz on $\mathcal{G}^A$, RGB on $\mathcal{G}$, and SAX* on the corresponding time series of occurrences of individual entities. We varied window size $|W|$ (starting in the given date), maximum size N of each output subgraph, and maximum number K of output subgraphs as follows:

- $|W| \in \{1, 2, 3, 4, 5\}$;
- $N \in \{5, 10, 15\}$;
- $K \in \{10, 15, 20\}$.

This led to a total of 45 different configurations to be given as input to Buzz and RGB. In the case of SAX*, instead, the only parameter that is defined is the window size $|W|$. Indeed, this algorithm allows for specifying neither the number N of stories nor the story size K. As in previous evaluation, for a given value of N and K, we retained the SAX* stories with size no more than N, and, if SAX* had output more than K stories, we sampled a random subset of size K. For the sake of robustness, the sampling procedure was repeated 10 times and performance indicators were obtained by averaging across the 10 samples.

4.4. Evaluation: Correspondence with real-world events

The overwhelming majority of methods that have been designed and developed to detect events in social-media and news data streams adopt the definition of an event introduced by research on Topic Detection and Tracking (TDT), i.e., a real-world occurrence that takes place in a certain geographical location and over a certain time period.

Evaluating whether and to what extent a method is able to detect real-world events is a difficult problem [91], due to the complex and subjective nature of the task, and to the lack of public annotated datasets and standard evaluation measures. Most of existing approaches [1,3,10,14,48,84,90,91,97] analyze very limited datasets, and either perform a manual evaluation of the events output by their method, or compare to a ground-truth list of events.

In our first experiment on search-log data (see previous section) we chose the first option, and we faced all the well-known difficulties that are typically encountered in editorial evaluation: the task was complex, subjective, and time consuming. For such reasons we decide to follow the second idea in this second experiment, and to build a ground truth of events that we can use to automatize the assessment of performance of Buzz and the competitors.

As said before, finding a ground truth is also a non-trivial issue. Very few corpora are publicly available [3,59] and, to the best of our knowledge, none is available spanning the very recent time interval covered by our dataset. Thus we built a ground truth on our own. In the recent literature, such a list of reference events has been built in various ways, for example, using hashtags [1], or fastest growing threads in a stream of twitter messages [14], or the top headlines provided by newspapers and news services [90,91]. We followed the last idea, which seems to be the most general and proper one given the nature of our input sources (newspapers, news portals, and social-media sites).

Various major international newspapers and news agencies provide API services that allow to retrieve the top news for a given date, and to refine the search by specifying query terms for a topic of interest, or

Table 11

News evaluation: results of the comparison to a ground truth of real-world events.

		Precision			Recall			F-measure			time (ms)		
Parameter		**RGB**	**SAX***	**Buzz**	**RGB**	**SAX***	**Buzz**	**RGB**	**SAX***	**Buzz**	**RGB**	**SAX***	**Buzz**
$\|W\|$	1	0.194	0.146	0.086	0.077	0.028	0.053	0.091	0.045	0.062	1 531	2 233	1 758
	2	0.194	0.119	0.156	0.077	0.032	0.086	0.091	0.045	0.099	1 531	2 374	3 806
	3	0.194	0.111	0.284	0.077	0.058	0.110	0.091	0.050	0.130	1 531	2 456	6 030
	4	0.194	0.043	0.282	0.077	0.028	0.101	0.091	0.023	0.119	1 531	2 623	8 353
	5	0.194	0.013	0.319	0.077	0.009	0.096	0.091	0.008	0.114	1 531	2 690	9 956
N	5	0.194	0.146	0.319	0.037	0.058	0.061	0.060	0.050	0.097	1 555	2 455	6 619
	10	0.160	0.146	0.236	0.059	0.058	0.089	0.080	0.050	0.120	1 552	2 455	6 308
	15	0.136	0.146	0.188	0.077	0.058	0.110	0.091	0.050	0.130	1 486	2 455	6 094
K	10	0.122	0.088	0.210	0.055	0.028	0.091	0.066	0.033	0.104	1 020	1 637	4 227
	15	0.160	0.117	0.268	0.067	0.052	0.099	0.079	0.047	0.115	1 531	2 455	6 340
	20	0.194	0.146	0.319	0.077	0.058	0.110	0.091	0.050	0.130	2 041	3 273	8 454

additional filters such as geographical location. As our dataset consists of Italian articles and information items, we picked, as a well-known, and yet external (i.e., not included in our list of input sources) source, the Italian version of `euronews.com`.

We automatically queried `it.euronews.com` for each date included in our dataset, and parsed the result extracting the suggested headlines. The result obtained by querying a specific date typically contained headlines published in the required date, and, in some cases, a few headlines referring to other dates close to the one of interest. Thus, we merged the lists of headlines obtained from all the queries, obtaining a reference list of 524 headlines covering the whole time frame spanned by out dataset. The average number of headlines per date in the ground truth is 5 and the maximum is 15.

Given the entity-based representation adopted to build our temporal graph, the events automatically extracted by Buzz and the competitors consist of a date, a temporal window and a set of entities (contained in the subgraph extracted by an algorithm). Thus, to make our results comparable with the ground truth, we also needed to build an entity-based representation of the news contained in the reference list obtained from `it.euronews.com`. We built such entity representations by employing the same NLP tool that we used to construct the news dataset, i.e., *Hermes* [19]. In particular, here we exploited the Hermes' crawling functionality to retrieve all the headlines, as well as its entity recognition and disambiguation module to tag such headlines with relevant Wikipedia entities mentioned in their content.

Results. We compared the results returned by Buzz, SAX*, and RGB, for all the parameter configurations listed before, to the events contained in the ground truth for the 24 selected test dates. Following the approach suggested by [3,65] we performed automatic comparison with the ground truth. To decide if a detected event covers a reference event, we compared the corresponding sets of entities by means of standard information-retrieval measures: precision, recall, and F-measure.

Table 11 reports the results of this evaluation. Results shown in correspondence of a certain parameter and criterion refer to the best results achieved on that criterion by varying all other parameters within the ranges specified above. Results clearly attest the superiority of the proposed Buzz method with respect to both the competitors. In fact, apart from the case $|W| = 1$, which, as discussed in more detail in the evaluation on search-log data, is not really meaningful for our method, Buzz achieved values of precision, recall and F-measure evidently higher than both the competing methods, up to double values in most cases.

In the same table we also report the (average) running times of the three competing methods. All methods were able to identify stories in a few seconds, with SAX* being the most efficient method.

Finally, in Table 12 we show the count of how many times a given parameter configuration led to the best result for each one of the selected methods. The table shows that $K = 20$ is always the best choice. As far as the window size, $|W| = 3$ seems to be a good choice on average. Parameter N has instead the largest variabil-

Table 12

News evaluation: summary of best parameter configurations: for each method and parameter configuration, number of events for which that configuration led to the best result.

method	parameters	#events
Buzz	$\|W\|$=1, N=15, K=20	64
	$\|W\|$=3, N=15, K=20	50
	$\|W\|$=2, N=15, K=20	48
	$\|W\|$=4, N=15, K=20	37
	$\|W\|$=5, N=15, K=20	37
	$\|W\|$=2, N=5, K=20	36
	$\|W\|$=5, N=5, K=20	36
	$\|W\|$=3, N=10, K=20	35
	$\|W\|$=3, N=5, K=20	34
	$\|W\|$=4, N=10, K=20	34
	$\|W\|$=4, N=5, K=20	26
	$\|W\|$=5, N=10, K=20	26
	$\|W\|$=2, N=10, K=20	24
	$\|W\|$=1, N=5, K=20	19
	$\|W\|$=1, N=10, K=20	18
RGB	$\|W\|$=1, N=15, K=20	341
	$\|W\|$=1, N=5, K=20	123
	$\|W\|$=1, N=10, K=20	60
SAX*	$\|W\|$=1, N=15, K=20	379
	$\|W\|$=1, N=5, K=20	93
	$\|W\|$=1, N=10, K=20	52

ity. This parameter is quite sensitive to the type of the specific event to be detected. However, a good general choice is to set it equal to 10.

5. Related Work

Story identification. Detecting emerging events/stories from user-generated content has received considerable attention in the last years [37]. Existing approaches fall into two main categories. The first one includes graph-based approaches [1,12,24,26,71,73,74,92,18,95,98], while the second one comprises methods that retain objects with anomalous behavior in a specific time window, without relying on any co-association graph [32,41,46,54,80,81,88,96]. In this work we propose a novel approach that combines ideas from the two existing categories and extract cohesive subgraphs (stories) in an anomalous co-association graph.

Among the graph-based approaches, it is worth mentioning the method by Bansal *et al.* [12], which extracts relevant keywords from a set of blog posts and build a graph representing the co-association among those keywords. Stories are ultimately identified by extracting clusters of keywords that are cohesively connected in the graph. Chen *et al.* [24] devise a method targeted to heterogeneous networks, i.e., networks composed of objects of multiple types. The method is based on maximizing a nonparametric scan statistic over connected subgraphs of the underlying heterogeneous network, so as to identify events as network clusters optimizing such a statistic. Das Sarma *et al.* [26] identify stories as a set of highly-correlated entities in a graph depicting the dynamic relationships between pairs of entities deriving from a stream of user-generated content. Rayana and Akoglu [71] devise an ensemble approach that systematically selects the results to assemble in a fully unsupervised fashion. Rozenshtein *et al.* [73] focus on activity networks, and define an event as a subset of nodes in the network that are close to each other and have high activity levels. Activity level is measured in two alternative ways: either as the sum of distances among all pairs of the event nodes, or in terms of a tree-based minimum-distance compactness measure. Sarkas *et al.* [74] tackle the event-detection problem by looking for the strongest associations between entities mentioned in a document collection, where the strength of association between entities is again measured in terms of cohesiveness within a properly-defined entity co-association graph. Weng and Lee [92] propose an approach that analyzes a Twitter stream, and builds signals for individual words by applying wavelet analysis on the frequency-based raw signals of the words. It then filters trivial words out by looking at their signal autocorrelations. The remaining words are ultimately clustered to form events with a modularity-based graph-partitioning technique. Xiao *et al.* [95] identify events by extracting topically- and temporally-coherent subgraphs in a properly-defined interaction meta-graph. The notion of temporal coherence is defined based on the assumption that a real-world event is discussed frequently in a relatively short time span. Topical coherence instead follows the intuition that events correspond to trees capturing the information flow over the interaction meta-graph. Zhao *et al.* [98] detect events from click-through data, i.e., log data of web-search engines. This data is first segmented into a sequence of bipartite graphs based on the user-defined time granularity. Next, the sequence of bipartite graphs is transformed into a dual graph, where each node is a query-

page pair that is used to represent events. Ultimately, the problem of event detection is formulated as the problem of clustering such a dual graph.

Non-graph-based story-identification methods are based on the identification of sets of terms/entities exhibiting anomalous temporal evolution in isolation, and a-posteriori grouping them in accordance with their temporal profile. Stilo and Velardi [80,81] assign each term a time series, describing how anomalous (according to a specific anomaly-detection model) its level of occurrence at any time instant is, when compared to the normal level of the whole time horizon. Events are defined as clusters of terms exhibiting a similar temporal trend in their time series. Gunnemann *et al.* [32] employ a statistical model for detecting events by spotting significant frequency deviations of the words' frequency over time. The statistical process is complemented with an optimization algorithm to extract only non-redundant events. Kumar *et al.* [46] investigate the problem of event detection in the context of real-time Twitter streams, and devise a method that employs single-pass clustering and distance compression. Liang *et al.* [54] propose an iterative spatial-temporal mining algorithm employing a signal-processing approach. Spatial-temporal term occurrences are viewed as signals, which are cleaned up by applying noise filters that are specifically targeted to improve the quality of an event-extraction task from these signals. The iterative-mining algorithm clusters terms and generates new filters based on the results of clustering in an alternating fashion. Vosoughi and Roy [88] devise a semi-automatic approach to story identification in Twitter. The method is based on the intuition that tweets related to a story contain assertions of that story. The proposed method is a two-step one: a proper assertion-detection step is followed by a hierarchical-clustering process that groups tweets into stories. Zhang *et al.* [96] detect local events from geo-tagged tweet streams. The Zhang *et al.*'s method employs a measure that captures the geo-topic correlations among tweets, and identifies pivots in the query window based on such a measure. These pivots form a list of candidate events, which are ultimately screened by summarizing continuous tweet streams and comparing the pivots against historical activities.

An orthogonal problem to story identification is how to efficiently maintain stories by incremental updating [8]. Existing incremental strategies do not work for the novel method we propose. Studying how buzzing stories can be efficiently maintained is a non-trivial problem that we defer to future work.

Finally, effort in this area has also been devoted to related (but different) problems, such as event evolution tracking, i.e., monitoring the evolution pattern of events [47,55], entity evolution discovery, i.e., discovering evolutions of entities from a stream of textual documents [69], topic/meme tracking, i.e., monitoring the evolution of specific topics or short, distinctive phrases [30,50], story-context identification, i.e., building story contexts based on the correlation with other stories [48,97], story-link detection, i.e., given two stories, determine if they are related to each other (e.g., talk about the same topic) [62,76], event-timeline generation. i.e., creating a coherent timeline for an event of interest [6,29], trend analysis, i.e., performing analysis of what is trending at a given point in time [43,83].

Anomaly detection in temporal data. Anomaly detection (also known as outlier detection) in temporal data is the problem of identifying objects whose behavior throughout a certain temporal horizon significantly deviates from the behavior of other objects. Two main variants of the problem exist: (*i*) anomaly detection in a set of temporal sequences, i.e., detecting sequences in a given set that exhibit anomalous behavior with respect to other sequences in that set, and (*ii*) anomaly detection in a single temporal sequence, i.e., detecting outlying points/subsequences within the same temporal sequence. Several approaches have been proposed in the literature, including unsupervised discriminative approaches [72], unsupervised parametric approaches [22], or supervised approaches [53] for the former variant of the problem, and prediction models [35], profile similarity-based approaches [93], or deviant-detection approaches [61] for the latter variant. For a comprehensive survey on the topic please refer to [33]. Another problem that is worth mentioning here is the problem of change detection in dynamic networks, whose goal is to discover significant changes in the structure of a network that evolves over time [15,20,56]. This problem can be seen as a special type of anomaly detection in temporal data.

In this work we resort to anomaly detection in a single temporal sequence to devise the first step of the proposed approach to identifying buzzing stories. Particularly, we interpret the weights assigned to an edge of the input temporal graph as a temporal sequence, and we quantify the anomaly level of each point (weight value) in the sequence based on some anomaly-detection model. In our approach we use a model that trades off between simplicity and effective-

ness (see Section 2.1). Our proposal is however orthogonal to this body of research as any other more sophisticated anomaly model can be employed.

Dense-subgraph discovery. Extracting dense substructures from a large graph is a well-established problem. Generally speaking, such a problem requires to find a subgraph (or a number of subgraphs [11,25, 85,86]) of a given input graph that optimizes some notion of density. Many definitions of dense subgraph have been proposed, such as cliques, quasi-cliques, *k*-cores, *n*-clans, *k*-plexes, *n*-clubs [49].

A well-established density notion is the average degree. Due to its popularity, the corresponding problem of finding a subgraph that maximizes the average degree has been commonly referred to as the *densest-subgraph* problem. The densest subgraph can be identified in polynomial time [31], and approximated within a factor of $\frac{1}{2}$ in linear time [23]. More difficult (i.e., **NP**-hard) variants of the densest-subgraph problem include the *densest-k-subgraph* problem, which asks for a densest subgraph of k vertices [9], as well as the *densest-at-least-k-subgraph* problem and the *densest-at-most-k-subgraph* problem, which asks for a densest subgraph of size respectively no less and no more than k [7,44].

Another well-known notion of dense subgraph is the *k-core*, defined as the maximal subgraph where all vertices have degree at least k [75]. The notion of *k*-core (and the related *core decomposition*) has been widely used, e.g., to quantify the global position of a vertex in a complex network [75], or as a heuristic to maximum-clique finding [27], or as a proxy for betweenness centrality [34].

In this work we use dense-subgraph discovery as a tool for the second step of our approach to identifying buzzing stories. Particularly, we define a density measure suitable for temporal graphs and devise algorithms to extract dense subgraphs according to this density definition. Density notions for temporal graphs have also been introduced in [17,94]. However, those notions are not suitable for our context. Indeed, the notion by Bogdanov *et al.* [17] works only for graphs having binary edge weights (within $\{-1, 1\}$), while Wu *et al.* [94] define a notion of core decomposition for temporal graphs, which does not admit any time window of interest as input, as required by our task.

6. Conclusions

The problem of automatically identifying buzzing events from user-generated content has raised a lot of interest in the last few years. Existing approaches fall into two main categories: approaches that extract stories as cohesive substructures in a graph representing the strength of association between terms (or entities), and approaches that study the behavior of terms over time and identify stories by a-posteriori grouping terms exhibiting similar anomalous temporal trends.

In this work we advance the literature on story identification from user-generated content by proposing a novel two-step method which profitably combine the peculiarities of the two main existing approaches, thus also overcoming their limitations. We conduct an extensive experimentation on two datasets respectively extracted from a real-world web-search log, and from a news corpus. Results attest the superiority of our approach over existing methods.

In the future we plan to investigate how buzzing stories can be updated incrementally. We will also focus on other anomaly-detection models in the first step, different notions of cohesiveness in the second step, and how to extract overlapping subgraphs, to allow objects to appear simultaneously in different stories. Finally, we would like to investigate the connections between the problem of identifying buzzing stories and concept drift.

Acknowledgement. This work was supported in part by the MIUR under grant *"Dipartimenti di eccellenza 2018-2022"* of the *Department of Computer Science* of *Sapienza University*.

References

[1] Charu C. Aggarwal and Karthik Subbian. Event detection in social streams. In *SDM*, pages 624–635, 2012.

[2] James Allan, Victor Lavrenko, Daniella Malin, and Russell Swan. Detections, bounds, and timelines: UMass and TDT-3. In *TDT Workshop*, pages 167–174, 2000.

[3] Hind Almerekhi, Maram Hasanain, and Tamer Elsayed. EveTAR: A new test collection for event detection in Arabic tweets. In *SIGIR*, pages 689–692, 2016.

[4] Nasser Alsaedi, Pete Burnap, and Omer Rana. Identifying disruptive events from social media to enhance situational awareness. In *ASONAM*, pages 934–941, 2015.

[5] Nasser Alsaedi, Pete Burnap, and Omer Rana. Can we predict a riot? Disruptive event detection using Twitter. *TOIT*, 17(2):18:1–18:26, 2017.

[6] Tim Althoff, Xin Luna Dong, Kevin Murphy, Safa Alai, Van Dang, and Wei Zhang. TimeMachine: Timeline generation for knowledge-base entities. In *KDD*, pages 19–28, 2015.

[7] Reid Andersen and Kumar Chellapilla. Finding dense subgraphs with size bounds. In *WAW*, pages 25–37, 2009.

[8] Albert Angel, Nikos Sarkas, Nick Koudas, and Divesh Srivastava. Dense subgraph maintenance under streaming edge weight updates for real-time story identification. *PVLDB*, 5(6):574–585, 2012.

[9] Yuichi Asahiro, Refael Hassin, and Kazuo Iwama. Complexity of finding dense subgraphs. *Discr. Ap. Math.*, 121(1-3):15–26, 2002.

[10] Farzindar Atefeh and Wael Khreich. A survey of techniques for event detection in Twitter. *Comput. Intell.*, 31(1):132–164, 2015.

[11] Oana Denisa Balalau, Francesco Bonchi, T-H. Hubert Chan, Francesco Gullo, and Mauro Sozio. Finding subgraphs with maximum total density and limited overlap. In *WSDM*, pages 379–388, 2015.

[12] Nilesh Bansal, Fei Chiang, Nick Koudas, and Frank Wm. Tompa. Seeking stable clusters in the Blogosphere. In *VLDB*, pages 806–817, 2007.

[13] Vladimir Batagelj and Matjaz Zaveršnik. Fast algorithms for determining (generalized) core groups in social networks. *ADAC*, 5(2):129–145, 2011.

[14] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on Twitter. In *ICWSM*, 2011.

[15] Michele Berlingerio, Francesco Bonchi, Björn Bringmann, and Aristides Gionis. Mining graph evolution rules. In *ECML-PKDD*, pages 115–130, 2009.

[16] Petko Bogdanov, Ben Baumer, Prithwish Basu, Amotz Bar-Noy, and Ambuj K. Singh. As strong as the weakest link: Mining diverse cliques in weighted graphs. In *ECML-PKDD*, pages 525–540, 2013.

[17] Petko Bogdanov, Misael Mongiovì, and Ambuj K. Singh. Mining heavy subgraphs in time-evolving networks. In *IEEE ICDM*, pages 81–90, 2011.

[18] Francesco Bonchi, Ilaria Bordino, Francesco Gullo, and Giovanni Stilo. Identifying buzzing stories via anomalous temporal subgraph discovery. In *IEEE/WIC/ACM WI*, pages 161–168, 2016.

[19] Ilaria Bordino, Andrea Ferretti, Marco Firrincieli, Francesco Gullo, Marcello Paris, Stefano Pascolutti, and Gianluca Sabena. Advancing NLP via a distributed-messaging approach. In *IEEE Big Data*, pages 1561–1568, 2016.

[20] B. Bringmann, M. Berlingerio, F. Bonchi, and A. Gionis. Learning and predicting the evolution of social networks. *IEEE Intelligent Systems*, 25(4):26–35, 2010.

[21] Carlos Castillo. *Big Crisis Data: Social Media in Disasters and Time-Critical Situations*. Cambridge University Press, New York, NY, USA, 1st edition, 2016.

[22] Varun Chandola, Varun Mithal, and Vipin Kumar. Comparative evaluation of anomaly detection techniques for sequence data. In *IEEE ICDM*, pages 743–748, 2008.

[23] Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *APPROX*, pages 84–95, 2000.

[24] Feng Chen and Daniel B. Neill. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In *KDD*, pages 1166–1175, 2014.

[25] Jie Chen and Yousef Saad. Dense subgraph extraction with application to community detection. *TKDE*, 24(7):1216–1230, 2012.

[26] Anish Das Sarma, Alpa Jain, and Cong Yu. Dynamic relationship and event discovery. In *WSDM*, pages 207–216, 2011.

[27] David Eppstein, Maarten Löffler, and Darren Strash. Listing all maximal cliques in sparse graphs in near-optimal time. In *ISAAC*, pages 403–414, 2010.

[28] Paolo Ferragina and Ugo Scaiella. TAGME: on-the-fly annotation of short text fragments (by Wikipedia entities). In *CIKM*, pages 1625–1628, 2010.

[29] Johanna Fulda, Matthew Brehmer, and Tamara Munzner. TimeLineCurator: Interactive authoring of visual timelines from unstructured text. *IEEE TVCG*, 22(1):300–309, 2016.

[30] Zekai Gao, Yangqiu Song, Shixia Liu, Haixun Wang, Hao Wei, Yang Chen, and Weiwei Cui. Tracking and connecting topics via incremental hierarchical Dirichlet processes. In *IEEE ICDM*, pages 1056–1061, 2011.

[31] Andrew V. Goldberg. Finding a maximum density subgraph. Technical report, University of California at Berkeley, 1984.

[32] Nikou Günnemann and Jürgen Pfeffer. Finding non-redundant multi-word events on Twitter. In *ASONAM*, pages 520–525, 2015.

[33] Manish Gupta, Jing Gao, Charu C. Aggarwal, and Jiawei Han. Outlier detection for temporal data: A survey. *TKDE*, 26(9):2250–2267, 2014.

[34] John Healy, Jeannette Janssen, Evangelos E. Milios, and William Aiello. Characterization of graphs using degree cores. In *WAW*, pages 137–148, 2006.

[35] David J. Hill and Barbara S. Minsker. Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. *Environmental Modelling and Software*, 25(9):1014–1022, 2010.

[36] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *EMNLP*, pages 782–792, 2011.

[37] Yuheng Hu, Yu-Ru Lin, and Jiebo Luo. Collective sensemaking via social sensors: Extracting, profiling, analyzing, and predicting real-world events. In *KDD*, pages 2127–2128, 2016.

[38] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Processing social media messages in mass emergency: A survey. *ACM Comput. Surv.*, 47(4):67:1–67:38, June 2015.

[39] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. AIDR: Artificial intelligence for disaster response. In *WWW*, pages 159–162, 2014.

[40] Syed Tanveer Jishan, Md. Nuruddin Monsur, and Hafiz Abdur Rahman. Breaking news detection from the web documents through text mining and seasonality. *Int. J. Knowl. Web Intell.*, 5(3):190–207, 2016.

[41] Janani Kalyanam, Sumithra Velupillai, Mike Conway, and Gert R. G. Lanckriet. From event detection to storytelling on microblogs. In *ASONAM*, pages 437–442, 2016.

[42] Margarita Karkali, François Rousseau, Alexandros Ntoulas, and Michalis Vazirgiannis. *Efficient Online Novelty Detection in News Streams*, pages 57–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[43] Noriaki Kawamae. Trend analysis model: Trend consists of temporal words, topics, and timestamps. In *WSDM*, pages 317–326, 2011.

[44] Samir Khuller and Barna Saha. On finding dense subgraphs. In *ICALP*, pages 597–608, 2009.

[45] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of Wikipedia enti-

ties in web text. In *KDD*, pages 457–466, 2009.

[46] Shamanth Kumar, Huan Liu, Sameep Mehta, and L. Venkata Subramaniam. Exploring a scalable solution to identifying events in noisy Twitter streams. In *ASONAM*, 2015.

[47] Pei Lee, Laks V. S. Lakshmanan, and Evangelos E. Milios. Incremental cluster evolution tracking from highly dynamic network data. In *ICDE*, pages 3–14, 2014.

[48] Pei Lee, Laks V.S. Lakshmanan, and Evangelos Milios. CAST: A context-aware story-teller for streaming social content. In *CIKM*, pages 789–798, 2014.

[49] Victor E. Lee, Ning Ruan, Ruoming Jin, and Charu C. Aggarwal. A survey of algorithms for dense subgraph discovery. In *Managing and Mining Graph Data*. 2010.

[50] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD*, pages 497–506, 2009.

[51] Jianxin Li, Zhenying Tai, Richong Zhang, Weiren Yu, and Lu Liu. Online bursty event detection from microblog. In *UCC*, pages 865–870, 2014.

[52] Rui Li, Kin Hou Lei, Ravi Khadiwala, and Kevin Chen-Chuan Chang. TEDAS: A twitter-based event detection and analysis system. In *ICDE*, pages 1273–1276, 2012.

[53] Xiaolei Li, Jiawei Han, Sangkyum Kim, and Hector Gonzalez. ROAM: Rule- and motif-based anomaly detection in massive moving object data sets. In *SDM*, pages 273–284, 2007.

[54] Yuan Liang, James Caverlee, and Cheng Cao. A noise-filtering approach for spatio-temporal event detection in social media. In *ECIR*, pages 233–244, 2015.

[55] Hongyan Liu, Jun He, Yingqin Gu, Hui Xiong, and Xiaoyong Du. Detecting and tracking topics and events from web search logs. *TOIS*, 30(4):21:1–21:29, 2012.

[56] Corrado Loglisci, Michelangelo Ceci, and Donato Malerba. Relational mining for discovering changes in evolving networks. *Neurocomputing*, 150(PA):265–288, 2015.

[57] Xinjiang Lu, Zhiwen Yu, Bin Guo, Jiafan Zhang, Alvin Chin, Jilei Tian, and Yang Cao. Trending words based event detection in Sina Weibo. In *BigDataScience*, pages 4:1–4:6, 2014.

[58] Richard McCreadie, Craig Macdonald, Iadh Ounis, Miles Osborne, and Sasa Petrovic. Scalable distributed event detection for Twitter. In *IEEE BigData*, pages 543–549, 2013.

[59] Andrew J. McMinn, Yashar Moshfeghi, and Joemon M. Jose. Building a large-scale corpus for evaluating event detection on Twitter. In *CIKM*, pages 409–418, 2013.

[60] Rada Mihalcea and Andras Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *CIKM*, pages 233–242, 2007.

[61] S. Muthukrishnan, Rahul Shah, and Jeffrey Scott Vitter. Mining deviants in time series data streams. In *SSDBM*, pages 41–50, 2004.

[62] Tadashi Nomoto. Two-tier similarity model for story link detection. In *CIKM*, pages 789–798, 2010.

[63] Alexandra Olteanu, Sarah Vieweg, and Carlos Castillo. What to expect when the unexpected happens: Social media communications across crises. In *CSCW*, pages 994–1009, 2015.

[64] Miles Osborne, Saĺa Petrovic, Richard McCreadie, Craig Macdonald, and Iadh Ounis. Bieber no more: First story detection using Twitter and Wikipedia. In *TAIA*, 2012.

[65] Sasa Petrovic. *Real-time event detection in massive streams*. PhD thesis, University of Edinburgh, UK, 2013.

[66] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to Twitter. In *NAACL HLT*, pages 181–189, 2010.

[67] Sarah Pink, Kerstin Leder Mackley, Val Mitchell, Garrath T Wilson, and Tracy Bhamra. Refiguring digital interventions for energy demand reduction: Designing for life in the digital–material home. *Digital Materialities: Design and Anthropology*, page 79, 2016.

[68] Sarah Pink, Shanti Sumartojo, Deborah Lupton, and Christine Heyes La Bond. Mundane data: The routines, contingencies and accomplishments of digital living. *Big Data & Society*, 4(1), 2017.

[69] Gianvito Pio, Pasqua Fabiana Lanotte, Michelangelo Ceci, and Donato Malerba. Mining temporal evolution of entities in a stream of textual documents. In *ISMIS*, pages 50–60, 2014.

[70] Martin F. Porter. *An algorithm for suffix stripping*, pages 313–316. Morgan Kaufmann Publishers Inc., 1997.

[71] Shebuti Rayana and Leman Akoglu. Less is more: Building selective anomaly ensembles with application to event detection in temporal graphs. In *SDM*, pages 622–630, 2015.

[72] Umaa Rebbapragada, Pavlos Protopapas, Carla E. Brodley, and Charles R. Alcock. Finding anomalous periodic time series. *Machine Learning*, 74(3):281–313, 2009.

[73] Polina Rozenshtein, Aris Anagnostopoulos, Aristides Gionis, and Nikolaj Tatti. Event detection in activity networks. In *KDD*, pages 1176–1185, 2014.

[74] Nikos Sarkas, Albert Angel, Nick Koudas, and Divesh Srivastava. Efficient identification of coupled entities in document collections. In *ICDE*, pages 769–772, 2010.

[75] Stephen B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269–287, 1983.

[76] Chirag Shah, W. Bruce Croft, and David Jensen. Representing documents with named entities for story link detection (SLD). In *CIKM*, pages 868–869, 2006.

[77] Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *TKDE*, 27(2):443–460, 2015.

[78] Mauro Sozio and Aristides Gionis. The community-search problem and how to plan a successful cocktail party. In *KDD*, pages 939–948, 2010.

[79] Thomas Steiner, Seth van Hooland, and Ed Summers. MJ no more: Using concurrent Wikipedia edit spikes with social network plausibility checks for breaking news detection. In *WWW Companion*, pages 791–794, 2013.

[80] Giovanni Stilo and Velardi. Efficient temporal mining of micro-blog texts and its application to event discovery. *DAMI*, 30(2):372–402, 2016.

[81] Giovanni Stilo and Paola Velardi. Time makes sense: Event discovery in twitter using temporal similarity. In *IEEE/WIC/ACM WI*, pages 186–193, 2014.

[82] Melanie Swan. Sensor mania! The Internet of Things, wearable computing, objective metrics, and the quantified self 2.0. *Journal of Sensor and Actuator Networks*, 1(3):217–253, 2012.

[83] Xuning Tang and Christopher C. Yang. TUT: A statistical model for detecting trends, topics and user interests in social media. In *CIKM*, pages 972–981, 2012.

[84] Nicholas A. Thapen, Donal Stephen Simmie, and Chris Hankin. The early bird catches the term: Combining Twitter and news data for event detection and situational awareness. *CoRR*, abs/1504.02335, 2015.

[85] Charalampos Tsourakakis, Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Maria Tsiarli. Denser than the densest subgraph: Extracting optimal quasi-cliques with quality

guarantees. In *KDD*, pages 104–112, 2013.

[86] Elena Valari, Maria Kontaki, and Apostolos N. Papadopoulos. Discovery of top-k dense subgraphs in dynamic graph collections. In *SSDBM*, pages 213–230, 2012.

[87] Sarah Vieweg, Carlos Castillo, and Muhammad Imran. *Integrating Social Media Communications into the Rapid Assessment of Sudden Onset Disasters*, pages 444–461. Springer International Publishing, Cham, 2014.

[88] Soroush Vosoughi and Deb Roy. A semi-automatic method for efficient detection of stories on social media. In *ICWSM*, pages 707–710, 2016.

[89] Jeroen B.P. Vuurens and Arjen P. de Vries. First story detection using multiple nearest neighbors. In *SIGIR*, pages 845–848, 2016.

[90] Andreas Weiler, Michael Grossniklaus, and Marc H. Scholl. Evaluation measures for event detection techniques on Twitter data streams. In *BICOD*, pages 108–119, 2015.

[91] Andreas Weiler, Michael Grossniklaus, and Marc H. Scholl. Editorial: Survey and experimental analysis of event detection techniques for twitter. *The Computer Journal*, 2016.

[92] Jianshu Weng and Bu-Sung Lee. Event detection in Twitter. In *ICWSM*, 2011.

[93] Andrew W. Williams, Soila M. Pertet, and Priya Narasimhan. Tiresias: Black-box failure prediction in distributed systems. In *IPDPS*, pages 41–50, 2007.

[94] Huanhuan Wu, James Cheng, Yiping Ke, Yuzhen Huang, Da Yan, and Hejun Wu. Core decomposition in large temporal graphs. In *IEEE BigData*, pages 649–658, 2015.

[95] Han Xiao, Polina Rozenshtein, and Aristides Gionis. Discovering topically- and temporally-coherent events in interaction networks. In *ECML-PKDD*, pages 690–705, 2016.

[96] Chao Zhang, Guangyu Zhou, Quan Yuan, Honglei Zhuang, Yu Zheng, Lance Kaplan, Shaowen Wang, and Jiawei Han. GeoBurst: Real-time local event detection in geo-tagged tweet streams. In *SIGIR*, pages 513–522, 2016.

[97] Meng Zhao, Chen Zhang, Siyu Lu, and Hui Zhang. STeller: An approach for context-aware story detection using different similarity metrics and dense subgraph mining. In *CSCWD*, pages 152–157, 2016.

[98] Qiankun Zhao, Tie-Yan Liu, Sourav S. Bhowmick, and Wei-Ying Ma. Event detection from evolution of click-through data. In *KDD*, pages 484–493, 2006.