Identifying Buzzing Stories via Anomalous Temporal Subgraph Discovery

Francesco Bonchi
ISI Foundation
francesco.bonchi@isi.it

Ilaria Bordino
UniCredit, R&D Dept.
ibordino@acm.org

Francesco Gullo
UniCredit, R&D Dept.
gullof@acm.org

Giovanni Stilo
Sapienza University
stilo@di.uniroma1.it

***Abstract*—Story identification from online user-generated content has recently raised increasing attention. Existing approaches 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 limitations.**

In this work we advance the literature on story identification by devising a novel method that profitably combines the peculiarities of the two main existing approaches, thus also addressing their weaknesses. Experiments on a dataset extracted from a real-world web-search log demonstrate the superiority of the proposed method over the state of the art.

I. Introduction

The problem of automatically identifying stories or events[1] from online user-generated content has recently attracted a great deal of attention [1, 2, 7, 14, 16, 19, 21, 23]. Generally speaking, the goal is to take data from online sources, such as queries issued to a web search engine or posts from micro-blogging or 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 [2, 7, 16, 21, 23]. 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, 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 [19, 20]. 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, 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 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 (Section II).
- 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 (Section II-A).

[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 maximized and the corresponding combinatorial optimization problem. We show that the problem is **NP**-hard, and we devise an efficient and effective heuristic to solve it (Section II-B).
- We perform an extensive evaluation on a real dataset extracted from the query log of a popular search engine. Results 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 (Section III).

II. Anomalous Temporal Subgraph Discovery

Given a set of objects $\mathcal{O}$, a discrete time horizon $\mathcal{T}$, and a function $f : \mathcal{O} \times \mathcal{O} \times \mathcal{T} \to \mathbb{R}^+$ 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, or web search logs [1, 21, 23]. $\mathcal{T}$ 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 [2, 16].

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 fixed 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 $f_t : E_t \to \mathbb{R}^+$ 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.

In this work we study the problem of identifying *buzzing stories* from user-generated content. 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. Sections II-A and II-B 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 II-C summarizes the overall proposed approach.

A. 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 [9]. 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. 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 III. Our approach is however parametric to the anomaly-detection model: other models 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

Algorithm 1 AnomalyScores

Input: A temporal graph $\mathcal{G} = (V, \{E_t, f_t\}_{t \in \mathcal{T}})$, an integer r
Output: An anomalous temporal graph $\mathcal{G}^A = (V, \{E_t, \phi_t\}_{t \in \mathcal{T}})$

for all $t_i \in \mathcal{T}$ **do**
  $TOT(t_i) \leftarrow \sum_{e \in E_{t_i}} f_{t_i}(e)$
  for all $e \in E_{t_i}$, let $f'_{t_i}(e) := f_{t_i}(e)/TOT(t_i)$
  sort edges $e \in E_{t_i}$ by ascending $f'_{t_i}(e)$
  for all $e \in E_{t_i}$ following the order given by $f'_{t_i}(e)$ **do**
    $p_{t_i}(e) \leftarrow \sum_{e' \in E_{t_i} | f'_{t_i}(e') \leq f'_{t_i}(e)} f'_{t_i}(e')$
    $\phi_{t_i}(e) \leftarrow 0$
    if $i - r > 0 \ \wedge \ p_{t_i}(e) > p_{t_{i-r}}(e)$ **then**
      $\phi_{t_i}(e) \leftarrow p_{t_i}(e) - p_{t_{i-r}}(e)$

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}. As an example, if the input horizon $\mathcal{T}$ has a daily granularity, the reference r could be one week/month before. The final anomaly score assigned to e at time t_i is the difference between the two percentiles.

The pseudocode of our anomaly-detection model is reported as Algorithm 1. The time complexity of Algorithm 1 is $\mathcal{O}(|\mathcal{T}| m \log n)$, where $n = |V|$ and $m = \max_{t \in \mathcal{T}} |E_t|$.

B. 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:

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

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 $\mathcal{S}$:

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

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 [5, 18]), 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 [1] 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* $\mathbf{NP}$*-hard.*

Proof. We prove $\mathbf{NP}$-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 $\mathbf{NP}$-hard, we devise a fast heuristic that yields accurate solutions in practice, as confirmed by our experiments in Section III.

The proposed heuristic is inspired by the fact that a simplified version of our ATSD problem can be solved in polynomial time. Particularly, we consider a simplified ATSD that requires only one output subgraph ($K = 1$) and leaves the size of the output subgraph unbounded ($N = \infty$). We call this problem UNBOUNDED-ATSD (U-ATSD).

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 ATSD resembles the problem of finding the *inner-most core* in a graph (and the notion of *core decomposition*) [17], which, we briefly recall below.

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,

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) $S^* \subseteq V$.

$\mathbf{c} \leftarrow \emptyset, \quad \mathbf{Q} \leftarrow \emptyset$
for all $u \in V$ **do**
  $p(u) \leftarrow \min_{t \in W} deg(u,t)$
  insert u in $\mathbf{Q}$ with priority score $p(u)$
$k \leftarrow 0$
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**
    $k \leftarrow p(u)$
  $\mathbf{c}[u] \leftarrow k$
{update priority queue}
  for all $t \in W$, $v \in \mathbf{Q} \mid (u,v) \in E_t$ **do**
    $deg(v,t) \leftarrow deg(v,t) - \phi_t(u,v)$
  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 $p'(v) \neq p(v)$)
  remove u from $\mathcal{G}^A$
$S^* \leftarrow \{u \in V \mid \mathbf{c}[u] = k\}$

for all $k \in [1..k^*]$, forms the *core decomposition* of G. The linear time algorithm proposed by Batagelj and Zaveršnik [3] 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 minimum 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. The pseudocode of the U-ATS algorithm is reported as Algorithm 2. The time complexity of Algorithm 2 is $\mathcal{O}(|W| m \log n)$ ($n = |V|$, $m = \max_{t \in \mathcal{T}} |E_t|$). 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.

The next theorem shows 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 $\forall v \in V : g(v, C_1) \leq g(v, C_2)$ [3]. 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 [3]: 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 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 $\hat{S}_1 = S$
    $S \leftarrow \operatorname{argmax}_{i \in [|S|-N+1..|S|]} \delta(\hat{S}_i, W)$
  remove the subgraph induced by S from $\mathcal{G}^A$
  $\mathcal{S}^* \leftarrow \mathcal{S}^* \cup \{S\}$

Algorithm 4 Buzz

Input: A temporal graph $\mathcal{G} = (V, \{E_t, f_t\}_{t \in \mathcal{T}})$, a time window W, three integers $r, K, N \geq 1$.
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 $\mathcal{O}(K|W|m \log n)$.

C. 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 $\mathcal{O}((K|W|) \times m \log n)$.

III. Experimental evaluation

This section describes the empirical evaluation we conducted to assess the performance of our Buzz method.

Dataset. We experimented with real-world data extracted from a query log of a popular commercial web-search engine.[2]Web-search queries have been traditionally used in the story-identification literature [13, 23]. 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 II-A) with $r = 7$, i.e., setting the reference time for computing anomaly scores 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 [19, 20], 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 I reports statistics on $\mathcal{G}^A$ and $\mathcal{G}$.

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 [2, 7, 16, 21, 23]. 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* [19].

[2]Yahoo Web Search

TABLE I: Statistics for $\mathcal{G}$ and $\mathcal{G}^A$

	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

TABLE II: Examples of stories detected by our 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

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.

TABLE III: Examples of stories detected by the RGB baseline.

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

TABLE IV: Examples of stories detected by the SAX* baseline.

#	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

A. Anecdotal evidence

In Tables II–IV we show some examples of buzzing stories extracted by the proposed Buzz, and the RGB and SAX* baselines, respectively. Table II 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 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 III and IV 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 single one, likely due to the fact that SAX* does not admit any bound on the size of the output stories.

TABLE V: Search frequency of the buzzing stories extracted.

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

B. 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 [15], and sorting the stemmed terms lexicographically. For each story, we counted the number of distinct days it occurs in at least one query of that day, as well as mean frequency over all its daily occurrences. For each method, we then computed avg and max of the above counts over all buzzing stories and reported them in Table V. A striking difference exists among RGB on one side, and SAX* and Buzz on the other side. RGB finds stories corresponding to over-popular searches: half of the RGB's stories appear in the log almost every day (552 days over a

TABLE VI: 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

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 very 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, which not identify buzzing stories.

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 VI. 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.

C. 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 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

TABLE VII: Editorial evaluation

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

TABLE VIII: 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 %

provided us with 3 labels for each story. Each story was assigned the label that was chosen by at least two editors.

Table VII 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 [19]: 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 VIII 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 VIII 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: Buzz is significantly faster than RGB and slightly faster than SAX*.

IV. Related Work

Story identification. Existing approaches to story identification fall into two main categories. The first one includes graph-based approaches [2, 7, 16, 21, 23], while the second one comprises methods that retain objects with anomalous behavior in a specific time window, without relying on any co-association graph [19, 20]. 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. An orthogonal problem is how to efficiently maintain stories by incremental updating [1]. 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. Effort has also been devoted to related (but different) problems, such as event evolution tracking [11, 13, 14] or story-context identification [10].

Anomaly detection in temporal data. Anomaly detection in temporal data is the problem of identifying objects whose behavior in a temporal horizon deviates from the behavior of other objects [9]. In this work we resort to anomaly detection in the first step of our method, by employing a model that trades off between simplicity and effectiveness (see Section II-A). Our proposal is however orthogonal to this body of research as any other more sophisticated model can be used.

Dense-subgraph discovery. Extracting dense substructures from a graph is a well-established problem. Many definitions of dense subgraph exist [12], including the popular notion of average degree [6, 8] or *k-core* [17]. We use dense-subgraph discovery as a tool for the second step of our approach, by defining an appropriate definition of density for temporal graphs. Density notions for temporal graphs have also been introduced in [4, 22]. However, those notions are not suitable for our context. Indeed, the notion by Bogdanov et al. [4] is meaningful only for graphs that can have negative edge weights. If weights are all non-negative like in our setting, optimizing that density leads to trivial problem instances where the solution is given by the entire input graph. Wu et al. [22] instead define a notion of core decomposition for temporal graphs, which doesn't admit any time window of interest as input, as required by our task.

V. Conclusions

We have advanced 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. In the future we plan to investigate how buzzing stories can be updated incrementally. We will also focus on other user-generated content datasets, 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.

References

[1] A. Angel, N. Sarkas, N. Koudas, and D. Srivastava. Dense subgraph maintenance under streaming edge weight updates for real-time story identification. *PVLDB*, 5 (6):574–585, 2012.

[2] N. Bansal, F. Chiang, N. Koudas, and F. W. Tompa. Seeking stable clusters in the blogosphere. In *VLDB*, pages 806–817, 2007.

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

[4] P. Bogdanov, M. Mongiovì, and A. K. Singh. Mining heavy subgraphs in time-evolving networks. In *IEEE ICDM*, pages 81–90, 2011.

[5] P. Bogdanov, B. Baumer, P. Basu, A. Bar-Noy, and A. K. Singh. As strong as the weakest link: Mining diverse cliques in weighted graphs. In *ECML-PKDD*, 2013.

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

[7] A. Das Sarma, A. Jain, and C. Yu. Dynamic relationship and event discovery. In *WSDM*, pages 207–216, 2011.

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

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

[10] P. Lee, L. V. Lakshmanan, and E. Milios. Cast: A context-aware story-teller for streaming social content. In *CIKM*, pages 789–798, 2014.

[11] P. Lee, L. V. S. Lakshmanan, and E. E. Milios. Incremental cluster evolution tracking from highly dynamic network data. In *ICDE*, 2014.

[12] V. E. Lee, N. Ruan, R. Jin, and C. C. Aggarwal. A survey of algorithms for dense subgraph discovery. In *Managing and Mining Graph Data*. 2010.

[13] H. Liu, J. He, Y. Gu, H. Xiong, and X. Du. Detecting and tracking topics and events from web search logs. *TOIS*, 30(4):21:1–21:29, 2012.

[14] G. Pio, P. F. Lanotte, M. Ceci, and D. Malerba. *Mining Temporal Evolution of Entities in a Stream of Textual Documents*. Springer International, Cham, 2014.

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

[16] N. Sarkas, A. Angel, N. Koudas, and D. Srivastava. Efficient identification of coupled entities in document collections. In *ICDE*, 2010.

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

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

[19] G. Stilo and Velardi. Efficient temporal mining of micro-blog texts and its application to event discovery. *DAMI*, 2015.

[20] G. Stilo and P. Velardi. Time makes sense: Event discovery in twitter using temporal similarity. In *IEEE/WIC/ACM WI-IAT*, 2014.

[21] J. Weng, Y. Yao, E. Leonardi, and B. Lee. Event detection in Twitter. In *ICWSM*, 2011.

[22] h. Wu, J. Cheng, Y. Ke, Y. Huang, D. Yan, and H. Wu. Core decomposition in large temporal graphs. In *IEEE BigData*, 2015.

[23] Q. Zhao, T.-Y. Liu, S. S. Bhowmick, and W.-Y. Ma. Event detection from evolution of click-through data. In *KDD*, pages 484–493, 2006.