World Wide Web
DOI 10.1007/s11280-014-0301-5

User communities evolution in microblogs: A public awareness barometer for real world events

Maria Giatsoglou · Despoina Chatzakou · Athena Vakali

Received: 20 December 2013 / Revised: 11 June 2014 / Accepted: 22 June 2014

Abstract In social media, users' interactions are affected by real-world events which influence emergence and shifts of opinions and topics. Interactions around an event-related topic can be captured in a weighted network, while identification of connectivity and intensity patterns can improve understanding of users' interest on the topic. Community detection is studied here as a means to reveal groups of social media users with common interaction patterns in such networks. The proposed community detection approach identifies communities exploiting both structural properties and intensity patterns, while dynamics of communities' evolution around an event are revealed based on an iterative community detection and mapping scheme. We investigate the importance of considering interactions' intensity for community detection via a benchmarking process on synthetic graphs and propose a generic framework for: i) modeling user interactions, ii) identifying static and evolving communities around events, iii) extracting quantitative and qualitative measurements from the communities' timeline, iv) leveraging measurements to understand the events' impact. Two real-world case studies based on Twitter interactions demonstrate the framework's potential for capturing and interpreting associations among communities and events.

Keywords Community detection · Evolving community detection · Event tracking · User weighted interaction networks

M. Giatsoglou (✉) · D. Chatzakou · A. Vakali
Informatics Department, Aristotle University of Thessaloniki, Thessaloniki, Greece
e-mail: mgiatsog@csd.auth.gr

D. Chatzakou
e-mail: deppych@csd.auth.gr

A. Vakali
e-mail: avakali@csd.auth.gr

Published online: 18 July 2014

1 Introduction

In emerging social media, and particularly in microblogging, users' activity is usually tied to real world events, since users tend to report relevant information, or discuss their views on them through their posts, as the events unfold. At the same time, they *interact* in different ways, such as by replying to posts, referencing one another, etc. Such interactions in social media can be best captured in a *user interaction network* modelling users as nodes and their interactions as edges. A *user interaction network* can be viewed as a collection of entities whose interconnections are dynamic and multi-valued (since their intensities and types evolve over time). Such networks' analysis may demand different, more focused models, depending on the individual needs, thus often static network *snapshots* are extracted to facilitate the analysis task at hand. A snapshot, here, corresponds to the interactions observed between users within a given *time-frame* (i.e. a confined time-period), while variation in users' interaction frequency or intensity within this period can be captured by the assignment of different *weights* to the networks' associations, resulting in a weighted network. A time-aware analysis of such networks' structure by examining successive snapshots can reveal different patterns and dynamics, since users' behavior fluctuates depending on factors such as the co-existence of real-world events and users' interest in them, the emergence of users as key-actors in the events, the type of the social medium itself, etc. At the same time, topic- and event-specific networks can be derived from broader interaction networks by keeping as edges only interactions relevant to the given topic/event. Thus, in a different perspective, a per-snapshot analysis of an event-specific interaction network enables specifically focusing on communities which are relevant to the event's individual instances/phases.

Community detection targets at identifying such implicit groups in real world networks, with a community being generally defined on a graph, as: *a group of entities more densely connected to each other compared to the rest of the network*, and usually sharing common properties. Community detection has been applied in the context of social interaction networks, such as scientific co-authorship networks [2], corporate communication networks [27], and recently in social media user networks [30] [12]. The temporal aspect of communities' evolution is a more recent research problem which focuses on both detecting latent communities of entities, as well as tracking their evolution, by modelling and analyzing associations over time [10].

In this work, we study the reciprocities among real world events and relevant social media user communities, while we focus on revealing the form of communities generated with respect to events, opinions expressed within them, and changes they undergo as the event unfolds. Our main hypothesis is that by uncovering *user communities* and *community evolution chains*, and studying their quantitative and qualitative features in a *static* and *evolving* approach, respectively, we can better understand how users are affected by events and in what ways they embrace them. Apart from the societal conclusions that can emerge from such analysis, its application on targeted networks can be leveraged in domains such as: marketing (e.g. for estimating campaigns' success, or brand monitoring), politics and public affairs (e.g. for estimating the approval/disapproval of a new policy, or tracking uprisings of groups of people), news media and event organizing (e.g. for summarizing events based on the evolution of the public opinion as expressed in different communities).

Here, we present a generic framework for community evolution tracking in the spotlight of certain events of different characteristics, and elaborate on its successive steps. The proposed framework foresees community detection at different granularities and supports the tracking of communities across successive temporal network snapshots. This paper extends our previous approach in [9], which focuses on static interaction networks in the context of a real world event and proposes a community detection algorithm incorporating user interactions' intensities, while it projects the community analysis' results on the aggregated network on the time, topic and size axes to gain insights on the event's impact. Moving beyond [9], the framework proposed here can be applied on (time) evolving social media users interaction networks to extract finer grained dynamics of users interest on a given event, based on the community detection results. Apart from supporting the so called global event analysis, applied on the results of static community detection, the new framework is capable of identifying evolving community chains and leveraging them for the more detailed *intra-event* and *inter-event* analysis. Building on the idea that user interaction strengths are crucial in communities formation and that the communities detection and qualitative characterization can lead to a better understanding of real world events societal impact, we propose a frameworks instantiation via the community detection algorithm proposed in [9], which is suitable for user interaction networks.

Communities are initially detected in the context of a given time-frame, while they can, next, be analyzed at a community, time-frame, or time-frame group level. The proposed meta-analysis framework considers a set of features that can be extracted and analyzed at each granularity level. The association of the proposed community meta-analysis approach with the event impact tracking task is a contribution of this work, and can be leveraged for answering questions such as "*what types of communities are generated with respect to certain events*" and "*how does users' interest drift with respect to a given event*". We demonstrate the potential of the proposed approach by applying our framework on two real world user interaction datasets from Twitter, focused on events of different scales and characteristics. In specific, our focus here is on i) topics that are discussed in multiple, more focused event *instances* sharing a common context, and ii) one-time events that comprise targeted *sessions*, while for their tracking we propose the following analysis approaches:

- *global event* analysis that adopts the per community analysis approach to asses users' interest in an event theme as a whole, based on all observed user reactions;
- *intra-event* temporal analysis in terms of communities evolution, which exposes the (expansion versus shrinkage) forces characterizing a given event's impact on social media users, based on the per time-frame analysis approach;
- *cross-event* analysis that aligns *instances/sessions* with the appropriate time-frame groups to reveal the differences in users' behavior while shifting from one event *instance/session* to the next, given the general topic context.

The remainder of the paper is organized as next. Section 2 reviews literature on social media analysis for event tracking and (evolving) community detection in social networks. Section 3 presents our framework and sets up the problem, while Section 4 introduces and validates the proposed community detection approach. Section 5 presents experimentation on selected case studies and Section 6 concludes the paper.

2 Related work

Evolving social data mining has been applied for detecting non-obvious patterns in social interactions that indicate the coexistence of prominent events in the real world or in the context of the given social environment, often in real time. In social data mining, events have been defined as *occasions taking place at a specific time and location, e.g. concerts, parties* [32], or as *the information flow between a group of social actors on a specific topic* *over a certain time period* [24]. Both definitions use the concept of time, though, the first one refers to special events occurring at a real world location, whereas the second is based on social behavior focusing on users' activities over a specific topic. There are several event detection approaches, however here we focus on event tracking which can be applied when the given event is known.

Event tracking in social media Event tracking can be used for monitoring the evolution of a known event across time quantitatively and/or qualitatively. An event tracking system is proposed in [18], which detects peaks on the tweets' volume to identify moments of bursting users' activity, and then annotates the detected peaks and performs sentiment-based analysis on them. A probabilistic model is used in [16] to track events in terms of topic drift and users' interest, using both term-based frequencies and users' affiliation network, but without leveraging users' observed interactions. Conversations in Twitter, as they uncoil via the use of the *mentioning* mechanism in parallel to a specific event, is studied in [26]. Authors mention that such conversations could be used for revealing trends related to aspects of the given event/topic within user subgroups. The participation of Twitter users in discussions about various events based on their *type* (e.g. organizations, journalists, ordinary users) was studied in [6], where authors focused on identifying differences in users' participation in terms of the event's type and the types of the generated content.

Community detection in social media Community detection (or graph clustering) methods are diverse and can be classified based mainly on their definitions of communities and their algorithmic approaches in: i) cohesive subgraph discovery, ii) vertex clustering, iii) community quality optimization, iv) divisive, and v) model-based [23]. Although preliminary community detection algorithms were designed for small-scale networks with prominent community separation and had high complexity, recent approaches focus on complex large-scale graphs (such as web social interaction networks) and reveal communities that better match real world characteristics (e.g. [14, 15, 31]). In [32] event detection has been tackled by applying N-cut graph partitioning on two graphs: one connecting blog posts by their textual similarity, and another connecting pairs of users based on the similarity of their temporal activity profiles. By applying community detection on event-focused user interaction networks, we aim to identify latent groups of users that interact more often compared to the rest of the network and are focused on certain sub-topics of the event.

So far, few works have tackled community detection in social media users' interaction networks [12, 19]. In the context of social media, community detection has been mainly applied to friendship networks generated by the declared users' affiliations [11], resulting in easily-interpreted groups of users. Interaction networks, though, are more complex and their derived communities' interpretations are less obvious, since interactions among their members may indicate both awareness of each other as well as interest in common topics. Important aspects of community detection in social media are covered in [23] where the need to detect meaningful communities and also identify *hubs* and *outliers* is highlighted. This requirement is addressed in the local community detection algorithm SCAN [31], which also has an $O(N)$ time complexity with respect to the number of nodes. Up to now SCAN's applicability to weighted networks has only been addressed in [28] where a relevant structural similarity measure is proposed, but no explicit experimental results are offered for such networks. Thus, all previous efforts based on SCAN and its variants have been tested on limited (unweighted) synthetic or *closed-world* networks. *Closed-world* networks are limited within the scope of a certain domain (e.g. the Enron email network with internal company email exchanges), when on the contrary, social media users' interactions are of a wider scope and open nature, "connecting" people of different disciplines. Also, although weighted networks are a natural representation of user interactions' intensities, many community detection approaches, operate on unweighted networks after preserving either all relationships, or only those whose intensity exceeds a cut-off threshold. Here, in Section 4, we examine whether SCAN approaches can successfully uncover the underlying community structure in real world networks, or they need to be adapted to leverage the interactions' intensity.

Evolving community detection in social media Social network users' interactions vary in type, frequency and strength, thus the dynamics of communities' structure inevitably change over time. Different approaches have been suggested to capture communities evolution given the underlying network changes in [10]. Evolution in online forums is detected in [19] via an evolutionary clustering approach, whereas different types of interaction types are addressed jointly and separately in [17] and [11], respectively. In [29] several approaches for communities' evolution detection at different time-frames are presented, while in [22] statistical analysis on the detected evolving communities reveals correlations between their life span and size.

3 Community detection for event tracking in social media: Principles and guidelines towards a framework design

Communities evolution tracking in relation to a given event involves several decisions and a series of steps, namely: *event profiling*, *data collection and model formation*, *data mining*, *event & community co-analysis*. In the next paragraphs we will describe a generic framework for tracking the evolution of such communities and discuss its primary steps in detail. Table 1 summarizes the notation used throughout the paper.

3.1 Event profiling and analysis requirements specification

Event profiling is necessary since events differ in nature and in their inherent features. We focus on scheduled events that are either *recurring*, meaning that multiple event *instances* take place at different times under the umbrella of a general event theme, or *modular* that can be broken down in smaller *sessions* with a more focused theme.

Table 1 Notation used in the description of the framework

symbol	description
tf	time-frame: time window for data aggregation defined as a fixed number of time-units
tu	time-unit: defines how often the tf should be "shifted"; equals to the shifting step's length
$length(tf)$	time-frame's length: the number of tu that comprise a tf
tf_i	the i^{st} time-frame
$Int_i^{type}(u,v)$	the number of $type$ interactions observed during tf_i between users u and v
$Intensity_i(u,v)$	the overall intensity of interactions observed during tf_i between users u and v
α_{type}	the weight assigned to $type$ interaction for calculating $Intensity_i(u,v)$; $\alpha_{type} \in [0,1]$
$act_t^{type}(u,v)$	activity indicator: is 1 if u and v have a $type$ interaction at timestamp t, or 0 otherwise

Definition 1 A **recurring event** can be viewed as a series of event instances, with each instance having typically a daily *duration* and a sparse *instance granularity*, i.e. significant periods of time elapse between two successive event instances.

Definition 2 A **modular event** consists of sessions with a typical *duration* of a few hours and a dense *granularity*, i.e. sessions are consecutive.

A series of Eurogroup[1] meetings and a local TEDx[2] event can be viewed as examples of a recurring and a modular event, respectively, and are discussed in detail in Section 5. Apart from specifying the instance/session duration and granularity, when profiling an event additional features should be taken into consideration, such as the event's: *general topic(s)*, *impact scope*, and *intended audience*.

To track the impact of a given event, either *global communities* or *evolving community chains* (i.e. *evolving communities*) can be detected, depending on the **analysis requirements**. Global communities correspond to the interaction graph embedding all interactions observed with respect to the event, thus they are derived via static community detection, and they offer a general estimation about the total interest on the event, as well as on how this interest is expressed in different groups of people. If the event is relatively popular, there is a large number of interactions, and *global* communities are usually bigger and denser, compared to when selecting a graph snapshot spanning a shorter time duration. On the other hand, by co-analyzing evolving community chains, which are detected across successive graph snapshots, in parallel with the event's unfolding more fine-grained insights can be derived, such as local interest fluctuations, and more specific expressions of interest.

The events' profile and analysis requirements assist the selection of the framework's parameters. An essential parameter is the *keyword set*, which includes terms/ phrases related to the events' topic, while, optionally, a specific observation period and/or geolocation boundaries can be set for the event. When tracking evolving community chains, an important parameter is the *time-frame's* length ($length(tf)$), which specifies the temporal granularity of the data mining triggering, i.e. the frequency of the community detection's invocation, and the *time-unit* (tu), which represents how often and for how much the tf should be "shifted", under a sliding window approach. tu also regulates the overlap between two consecutive time-frames. In static (global) community detection, the $length(tf)$ parameter coincides with the duration of the observation period, while the tu parameter' granularity is irrelevant.

3.2 Data collection and model formation

Data are collected from a selected pool of social media and content is processed to extract users' interactions. Social media's selection could be based e.g. on the geographic focus in case of local events, along with the preferred social media of locals (e.g. a concert in China is discussed more in Weibo compared to Twitter), or even the intended events' audience (e.g. scientists, artists, etc). Microblogging sites are often used for event tracking since they

[1] Eurogroup is a forum of the finance ministers of the *eurozone*, i.e. European Union's member states whose currency is the euro, which aims at coordinating economic policies within eurozone and promoting conditions for economic growth. (http://www.eurozone.europa.eu/eurogroup/)

[2] TEDx events are independently planned and coordinated events that bring together local communities, organizations and individuals for presenting motivated and innovative ideas and engaging in dialogue. (http://www.ted.com/tedx/)

are considered a natural medium for sharing news/opinions and are used by users of various disciplines. Each web social platform supports different *types* of interactions, while the types of considered interactions, may lead to user networks of varying form/structure, and thus to different communities. In Twitter, e.g., explicit user interactions can be categorized in: *mentions*, which are usually made when referring to a user, *replies*, which are used for direct communication, and *retweets*, which are used for propagating the content generated by a user. To track communities evolution in dynamic event-driven networks, we generate and maintain an *evolving graph* of users under a *time-frame based* updating approach.

For evolving community detection, given the selected tu and $length(tf)$ (defined as a fixed number of time-units), we aggregate data under a moving window scheme, where the fixed-width tf moves forward one tu at a time. For each time-frame tf_i, we extract the observed interactions and update the user's interaction network by connecting users that have interacted at least in one tu within the tf's span. In Twitter e.g., interactions between users A and B that lead to their connection involve activities such as: A mentioning B in her post, A replying to or reposting (i.e. retweeting) a post of B. An edge between nodes u and v at a given tf_i, thus, represents the existence of an *aggregated interaction* between them, weighted by: $w_i(u,v) = \sum_{\forall type} \alpha_{type} \times Int_i^{type}(u,v)$, where $Int_i^{type}(u,v) = \sum_{t \in tf_i} act_t^{type}(u,v)$ stands for the sum of the observed $type$ interactions ($act_t^{type}(u,v) = 1$, if u and v have a $type$ interaction at timestamp t, or 0 otherwise). α_{type} is a weighting factor assigned to the interaction $type$. Due to our overlapping time-frames, for a given dataset we end up with a series of smoothly evolving networks encoding both the new tu's interactions and the interactions of the last $length(tf) - 1$ time-units. To achieve this, when the end of a tf_i is reached, the graph is updated by removing interactions observed during the tf's older tu (by decreasing accordingly the respective edges' intensity). Alternative interaction networks can be generated depending on the selection of participating interaction types and their weights, while all interaction types can be assigned a weighting factor equal to 1, if there is no need to assign different significance to each of them (this approach is also followed here in our experiments).[3] The approach described above is also followed in the simpler case of global (static) community detection, where there is just one tf covering the event's total monitoring period.

3.3 Data mining

Here, the selected community detection algorithm is applied on the graph's snapshot at the end of each tf, while a community matching cross time-frame approach is followed to determine each community's evolution. This approach to evolving community detection is characterized as *sequential mapping-driven evolving clustering* [10] and, in principle, it maps communities of each tf to their predecessors and successors (in the previous and following time-frames) with use of similarity measures and/or temporal smoothing techniques. Its focus is on detecting transitions in communities' life (birth, merge/split, death). The application of sequential mapping-driven evolving clustering on graphs generated via the previously described overlapping time-windowing process is proposed to alleviate noise and lead to a *smooth* community evolution, by refining abrupt changes in the monitored community structures.

[3] Interaction weights have been included in the framework in a simplified way, for generality's sake. In reality, estimating different weights for each interaction is difficult and a research problem on its own.

The framework's community detection module should implement an algorithm suitable for networks derived from web social activities, given that they are generally of large scale and present unpredictable bursts, while the resulting interaction networks can be rather complex, with communities difficult to discern. Also, in real world interaction networks, users should not necessarily be assigned to a community (a practice followed in graph partitioning), but they may either be considered as outliers weakly connected to the rest of the network, or hubs interconnecting several different communities of users. A user may be assigned to multiple communities, such as e.g. in a simple affiliation network, where two such communities could be the user's family and work circle. Thus, important features to be considered when selecting a community detection algorithm for the framework, depending on the needs of the problem at hand, include: i) low complexity, ii) support for overlapping communities, iii) distinction of user roles, iv) application on weighted networks, v) support for multiple interaction types, vi) parameter-free or having low sensitivity to parameters. In any case, the resulting communities should comprise nodes sharing similar *interaction patterns*, i.e. sets of *interaction instances*. Here, an interaction instance with respect to a given node (user) represents an *aggregated interaction* of a given weight between this node and a given neighbor.

To detect communities' evolution, we compare communities of successive time-frames tf_k and tf_{k+1} based on their node overlap (such as in [22]). If nodes similarity exceeds an overlap threshold, we consider that the given community in tf_{k+1} is the evolution of the matched community in tf_k and distinguish the transitions of: *growth*, *persistence*, *contraction*, depending on the change in the community's size. If a community in tf_k is matched to multiple communities in tf_{k+1}, we consider that it *splits* into them and then *dies*, whereas if multiple communities in tf_k are matched to one community in tf_{k+1} we consider it a merge. If no match is found for a community in tf_{k+1}, we consider that it has *died*. A *community chain* is, then, defined as follows:

Definition 3 A **community chain** is a series of communities detected in successive time-frames, constructed via the identification of its first appearance (*birth*), its *death*, and intermediate transitions of: *growth*, *persistence*, and *contraction*. The community chain's **life span** is the number of time-frames elapsed between its *birth* and *death*.

3.4 Event & community evolution co-analysis

Communities detected on each tf proceed to the *Event & Community Evolution Co-analysis* phase, in which, having each time-frame's communities at hand, the proposed framework extracts quantitative features characterizing communities on different granularity levels. Table 2 presents a selection of features that can be used to characterize the community detection results, along with aggregation operators that can be used on them at the three envisioned granularity levels (per community, time-frame, and time-frame group). In principle, community level analysis studies individual communities within the same tf in dimensions such as the communities' size, topic diversity and time span; time-frame level analysis gathers communities of the same tf and calculates aggregate features summarizing the time-frame's community set as a whole, while emphasis is on revealing changes observed over time; similarly, time-frame group level analysis operates on a number of successive time-frames and calculates their aggregate features, useful for comparing different time-frame groups.

Table 2 Community features per granularity level

Feature	Operators	Measured on
Size of community	–	community
	min, max, mean, std, sum	time-frame
Topic diversity in community	–	community
	min, max, mean, std	time-frame
Time span of community (time-units of activity)	–	community
	min, max, mean, std	time-frame
Number of communities	–	time-frame
Percentage of users in communities	–	time-frame
Size of community chain	–	community (chain)
	min, max, mean, std,	time-frame
	min, max, mean, std	time-frame group
Life span of community chain	–	community (chain)
	min, max, mean, std	time-frame
	min, max, mean, std	time-frame group
Number of community chains	–	time-frame group
Percentage of users in community chains	–	time-frame group

Global community detection combined with per community analysis is probably the simplest approach to follow for acquiring an initial, general impression on users' interest in an event theme as a whole. The total monitoring time period is taken as one (the only) time-frame, communities are detected on the induced graph, and the following features can be used to assist the interpretation of users' networking activities and interest with respect to the event [9]:

Definition 4 Given a community c detected at time-frame tf; a group of users U, where $\forall u_i \in U$ is member of c; a set of interactive, timestamped Twitter posts P, where $\forall p_k^t \in P$ embeds an interaction between users $u_i, u_j \in U$ which took place at time t; and a set of topics T expressed in P, the following features are defined:

- The **size of community** corresponds to the number of users assigned to c, $\mid U \mid$, and is indicative of its *strength* (in terms of popularity);
- The **topic diversity in community** is the size of the topic set of c, $\mid T \mid$;
- The **time span of community**, constrained in tf, is the length of the absolute temporal duration covered by its corresponding tweets: $t_{max} - t_{min}$, where $p_i^{t_{min}}, p_j^{t_{max}} \in P$, $t_{max} = \max_{p_k^t \in P} t$, $t_{min} = \min_{p_k^t \in P} t$.

To estimate the topics discussed in a particular time-frame, LDA [3], or a similar method, can be applied on the text of all tweets posted within it, and then tweets can be assigned to their most probable topic. A community's topic set T is then the union of all the topics expressed in the community's tweets. With this approach we can infer more refined topics that interest each community's members.

The evolving community detection setting is more suitable when, as described in Section 3.1, it is needed to track users' interest shifts across recurring and modular events.

To this end, our framework leverages the per time-frame and per time-frame group analysis approaches, described next, to perform *intra-event* and *inter-event* analysis, respectively. The per community analysis can still be useful when applied to comparatively analyze communities detected at a particular time-frame.

The *intra-event* analysis approach focuses on how a given event instance "evolves around itself" and can be applied to identify its evolution forces. Since depending on the instance's duration ($tu \times length(tf)$), its span may overlap with several time-frames, we align time-frames with all event instances/sessions to identify a relevant tf set for each of them. Based on a selected set of quantitative features, it is important to understand how their values evolve as the event unfolds from its beginning to its end, across these successive time-frames. To reveal these features' fluctuations and dynamics, a comprehensive visualization approach is needed to enable multiple features' comparison across the temporal dimension. The visualization approach followed should also be flexible, since it may need to accommodate multiple alternative "views" of the dataset in parallel, e.g. when multiple networks are generated under different interactions' combinations. Later, in our case studies' analysis, we present and demonstrate *EventWheels*, as an example of such a (dual) visualization. A category of features which can be generated at a per time-frame level involves the aggregation of the respective community level features, described above. Thus, a given tf can be characterized e.g. in terms of community's size by applying the statistical operators of Table 2 on the sizes of all relevant communities. For instance, the use of *mean community size* and *standard deviation of communities' size* per tf is demonstrated in our experimentation, as well as the sum of communities' size, which stands for the total *number of users in communities*, a feature that offers an indication of the overall interest expressed in the event at a given phase of time. Additional features that can be derived at a time-frame level are presented next:

- *number of communities*: it indicates the scale of the users' *dispersion* around the event. Dispersion is not necessarily thematic, since different communities may focus on similar topics. A community should rather be perceived as a group of users both interested in similar topics and sharing common interaction patterns;
- *percentage of users in communities*: it is the ratio of the number of all communities' members by the total number of users in the network (including outliers). This feature indicates how much clustered the users are at a given tf.

At a higher temporal granularity level, a *cross-event* analysis approach can be employed to compare the different event instances/sessions in their totality, instead of at a per tf basis, using both quantitative features, as well as some "event demographics". This approach groups all time-frames overlapping with a given instance/session, to derive measurements for it as a whole, and extracts features per time-frame group to use them for cross-event comparison. In this type of analysis, instead of communities, features are extracted in terms of community chains, since it is important to identify the appearance of the same community in successive time-frames. The features of the *number of community chains* and the *percentage of users in community chains* are derived for each event instance by taking the count of all different *community chains* that are "alive" in at least one relevant tf, and the union of all users participating to communities within a given tf set, similarly to the definitions provided for the intra-event analysis. The *mean lifespan* and *mean size* of a community chain (again it terms of users) are important features here, and are calculated over these "alive" community chains. The joint consideration of certain demographic features, such as e.g. the number of interactive tweets observed during the time period covered by the instance's tf

set, and their composition with respect to the interaction types can provide some additional context for the cross-event analysis.

4 Community detection for social media user interaction networks

To instantiate the framework, we propose an adaptation of SCAN [31], a community detection algorithm that builds on the density-based clustering algorithm DBSCAN [8]. While DBSCAN is widely used for clustering spatial points based on their density distribution, SCAN operates on graphs using a *structural similarity* measure. SCAN's advantages are that it is scalable, it identifies nodes acting as noise in the network (either outliers or hubs), and does not require the communities' number as input. It, however, does not support weights and its main limitation is its sensitivity to the selection of an initial similarity threshold parameter, whose fine-tuning requires repeated algorithm executions for several parameter values. An approach to alleviate SCAN's parameter sensitivity limitation was given in [28] employing the clustering quality *modularity* criterion [20] to find the optimal parameter's value. Alternative efforts addressing the parameter sensitivity of DBSCAN and SCAN led to the so called *reachability plots* [1, 4] which represent the algorithms' multiple clustering outcomes for all possible parameter combinations. A technique proposed in [25] operates on reachability plots produced by DBSCAN to automatically detect significant clusters.

Here, we examine if SCAN approaches can successfully uncover the underlying community structure in real world user interaction networks, or they need to be adapted to leverage the interactions' intensity. SCAN and the proposed adaptation for weighted networks WSCAN (i.e. WeightedSCAN) are evaluated in a series of synthetic networks. The combination of both approaches' experimental results with the corresponding intrinsic network properties (global *clustering* and *weighted clustering* coefficients [21]) leads to an empirical criterion for the selection of SCAN or WSCAN for the network at hand. WSCAN's limitation of parameter selection is also addressed by an automatic approach, AutoWSCAN, which detects communities from nodes' weighted structure connected order of traversal, inspired by [25].

4.1 Getting from SCAN to WSCAN

SCAN discovers cohesive network subclusters based on parameters μ and ε, controlling the minimum community's size and the minimum *structural similarity* between two community's nodes, respectively. Generally, a larger μ value leads to fewer and bigger communities, while a larger ε value to tighter communities and more outliers. With structural similarity as a clustering criterion, nodes with several common neighbors are placed in the same (μ, ε)-core community. To adapt SCAN for weighted networks we propose *weighted structure reachability* for (μ, ε)-cores' detection.

Definition 5 Given a weighted undirected network (G, w), where $G = \{V, E\}$ and $w : E \to \mathbb{R}$, the **weighted structural similarity** $wSSim$ of two nodes u and v is defined as:

$$wSSim(u, v) = \frac{\sum_{k\in\Gamma(u)\cap\Gamma(v)} w_{u,k} \cdot w_{v,k}}{\sqrt{\sum_{k\in\Gamma(u)} w_{u,k}^2}\sqrt{\sum_{k\in\Gamma(v)} w_{v,k}^2}} \ . \tag{1}$$

where $\Gamma(v)$ is the **neighborhood** of node v: $\Gamma(v) = \{k \in V|(v, k) \in E\} \cup \{v\}$, $w_{u,v} \in [0, 1)|u \neq v$; $w_{u,v} = 1|u = v$.

Definition 6 The **ε-neighborhood** of a given node u is the subset of its neighborhood containing only nodes that are at least ε-similar with u:

$$N_\varepsilon(u) = \{v \in \Gamma(u) | wSSim(u, v) \geq \varepsilon\} \ . \tag{2}$$

Definition 7 A vertex v is called a **(μ, ε)-core** if its ε-neighborhood contains at least μ vertices: $CORE_{\mu,\varepsilon}(v) \Leftrightarrow |N_\varepsilon(v)| \geq \mu$.

Additional nodes are attached to (μ, ε)-cores based on *structural connectivity*. Node u is *structure-reachable* from a core node v if u can be reached from v via a chain of nodes each belonging to the ε-neighborhood of the previous one. Nodes u and v are *structure-connected* if they are *reachable* from the same core node. A community is then defined as a set of structure-connected nodes that is maximal in terms of structure reachability. Nodes assigned to no community are characterized as either outliers or hubs depending on whether they are linked to a single or multiple communities, respectively. For calculating wSSim it is important to ensure that all weights are <1, since a weight of 1 is used as each node's self-similarity in the definition of wSSim. To achieve this, we scale all interactions' weights before community detection.

4.2 AutoWSCAN

Our experiments with WSCAN indicated its high sensitivity to parameter ε. Finding an ε value that leads to a balanced community structure regarding outliers' number, coherence, and communities' separation is, though, tedious. On the other hand, parameter μ affects SCAN-based algorithms only in terms of the minimum allowed cluster size, thus selecting a small μ value (e.g. 3-4) ensures that even very small sets of connected nodes are given the chance to become candidates communities. Due to the above, in this paper, we target our efforts only on the automatic selection of ε.

Xu et al. [31] proposes a heuristic approach for selecting ε based on the "knee-point hypothesis" for the μ-nearest neighbor similarity plot. Though, our application of this approach to real-world networks with both the "unweighted" and weighted structural similarity did not reveal clear knee-points. We thus adopt the structure connected order of traversal to represent the detected *structure-connected* community sets for all possible ε values [4]. AutoWSCAN (Alg. 1) re-orders nodes by structure-connected order of traversal based on *weighted core reachability* and *reachability similarity*.

Definition 8 Given a network (G, w), the **weighted core reachability** $wCSim$ of node u is defined as:

$$wCSim(u) = \begin{cases} wSSim(u, \mu NN(u)), if \ |\Gamma(u)| \geq \mu \\ UNDEFINED, else \end{cases} , \tag{3}$$

where $\mu NN(u)$ is the μ-nearest neighbor of node u.

Definition 9 Given a network (G, w), the **weighted reachability similarity** $wRSim$ of node v from node u is defined as:

$$wRSim(v, u) = \begin{cases} min \left(wCSim(u), wSSim(u, v)\right), if \ |\Gamma(u)| \geq \mu \\ UNDEFINED, else \end{cases} . \tag{4}$$

Algorithm 1 AutoWSCAN

Input: $G = (V, E, w)$, μ
Output: A sequence of nodes in structure-connected order of traversal.

```
foreach node v ∈ V do
    if v in visitedNodes then continue
    if visitedNodes is not empty then
        Cl=ClusterExtractor(orderedList)
        Communities.append(cl)
        orderedList.empty()
    enqueueNeighbors(v)
    if v is core then
        while visitQueue in not empty do
            currNode = visitQueue.getNode()
            enqueueNeighbors(curNode)
Function enqueueNeighbors (v)
  visitedNodes.add(v)
  orderedList.append(v)
  cs = computeWCoreSim(v)
  if v is core then
      foreach vN in v.neighbors do
          if visitedNodes not contains vN then
              ss = getWStructuralSim(v, vN)
              newWRSim = min(cs, ss)
              if vN.wReachSim is null then
                  vN.wReachSim = newWRSim
                  visitQueue.insert(vN, newWRSim)
              elif newWRSim > vN.wReachSim then
                  vN.wReachSim = newWRSim
                  visitQueue.setPriority(vN, newWRSim)
```

Weighted core reachability ($wRSim$) is calculated for each node, standing for the minimum ε value that would allow it to become a core (Alg. 1). Then, each possible core node u ($|\Gamma(u)| \geq \mu$) is "visited", which involves finding the node's neighbors, calculating their $wRSim$ from the current core, and inserting them at a priority queue based on the $wRSim$ value (or reordering the queue if they have already been inserted). At each iteration, the node with the highest $wRSim$ value from any previously visited node is extracted from the queue to ensure that regions of higher weighted structural similarity are spanned before surrounding areas of lower similarity [4]. The node visiting order represents the weighted structure connected order of traversal. For a connected network, Alg. 1 will never return to its first loop, though, since social media users' interaction networks are often disconnected, this is probable. Our approach is to generate partial nodes' sequences based on structure-connected order of traversal for each disconnected component and detect communities in them.

The weighted structure-connected order of traversal can be depicted in a *reachability plot*, which illustrates, in the corresponding order, the maximum weighted reachability value of each node from its previously visited nodes (i.e. $maxWRSim$). Figure 1 depicts a reachability plot, where we can observe areas in which the $maxWRSim$ values steadily rise and then fall at a local minimum to rise again after a while. Such "hills" represent different communities, whereas areas of low $maxWRSim$ values are outliers. Such communities can be revealed by 'slicing' the plot horizontally at a selected global similarity threshold, and isolating the regions that lay above it.

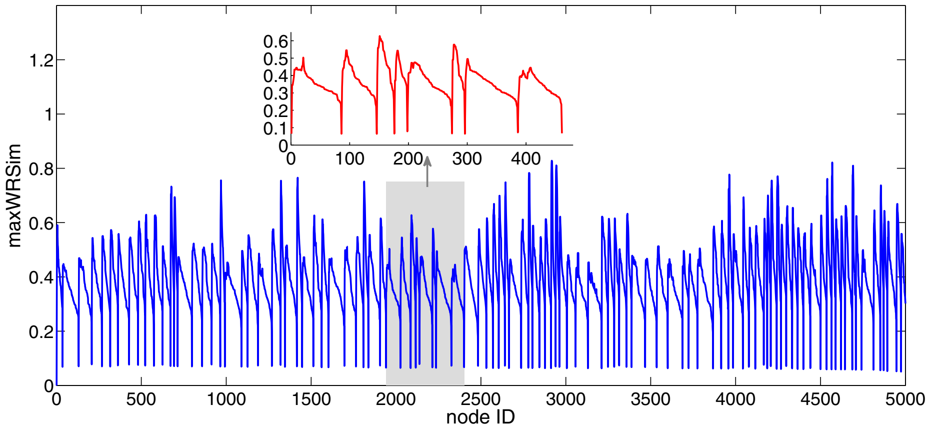


Figure 1 Example of a weighted reachability plot

Definition 10 Given a sequence of nodes $\{n_1, n_2 ..., n_{|V|}\}$ ordered based on weighted structure-connected order of traversal, a *community* is defined with respect to ε_{thres} as a subsequence of nodes $\{n_{a-1}, n_a, ..., n_b\}$ where $1 < a - 1 < b \leq |V|$, iff $\forall i \in [a, b]$, $maxWRSim(n_i) \geq \varepsilon_{thres}$ and [a,b] is maximal.

Algorithm 2 ClusterExtractor

Input: partial weighted reachabilityPlot: $WRPlot$, $minRatio$
Output: clusters

```
find localMinima in WRPlot
order localMinima from min to max
pNode.setRange(WRPlot(start), WRPlot(end))
return findClusters(pNode, localMinima)
Function findClusters (treeNode, localMinima)
  if localMinima is empty then
      if sizeOf(treeNode) > μ then
          treeNode is a cluster
      return
  lMin = localMinima.pop
  [leftNode, rightNode] = split(treeNode, lMin)
  remove all points before/after lMin having the same wRSim value
  if sizeOf(leftNode) > μ then leftNode: active
  if sizeOf(rightNode) > μ then rightNode: active
  if leftNode & rightNode inactive then
      treeNode is a cluster
  foreach activeNode do
      find its maximum wRSim_max value
      if (lMin/wRSim_max) > minRatio then
          ignore split point
          findClusters(treeNode, localMinima)
          return
      actMinima = localMinima in activeNode's range
      findClusters(activeNode, actMinima)
```

Since in real world networks communities are usually of different cohesion and strength, a global ε_{thres} will fail to identify communities of different *similarity-range* scales. Thus, to detect communities at different (local) ε values, we apply ClusterExtractor, an approach inspired by [25]. ClusterExtractor (Alg. 2) detects communities as contiguous areas between two local minima, satisfying some desired properties that reflect the way a person would identify communities by observing a reachability plot. It receives a weighted reachability plot and first identifies local minima points, ensuring that they have the lowest value in a subregion centered on them and spanning $2 \cdot \mu$ points. Then, it puts them in a priority queue by increasing value, and iteratively removes the first point from the queue and uses it to split the input nodes' sequence in two subregions. A split point is *valid* when the given subregions differ noticeably in their $maxWRSim$ values compared to the split's value. We, thus, check that the maximum value in each region is "significantly" larger than the split point's $maxWRSim$ (using a $minRatio \simeq 0.7$). ClusterExtractor is recursively called for each subregion whose size is larger than μ (active), and the same process is applied on subregions based on the minima points within their span. If there are no more (valid) minima points or both subregions are inactive, then the current region is a community.

4.2.1 Computational complexity

Given a graph with M edges and N nodes, the computational complexity of Alg. 1 for constructing the structured-connected order of traversal is dominated by the calculation of: $wRSim$ for all edges and $wCSim$ for all nodes (this entails finding each node's μ-nearest neighbor), as well as performing a single pass over all nodes through the 'visiting' process. The total cost of the above processes is: $O(M) + O(N * k) + O(N)$, where k is the node's average degree. Since many real world networks are sparse ($M \approx N$) and follow a power-law degree distribution (the degree's expected value is constant) [31], the average computational complexity for constructing the structured-connected order of traversal is $O(N)$. *ClusterExtractor* first involves finding the local minima of the input array and sorting them, a process which costs $O(N) + l * log(l)$, where l represents the number of local minima for a given input. Next, an additional pass over the input array allows to identify the points which are positioned next to the local minima and have the same value, which again takes $O(N)$. Each local minimum is examined for its suitability as a split point for either the whole array or some section of it (due to the algorithm's recursiveness) once, which demands, in the worst case, finding the maximum value of the whole array, thus imposing a total cost of $O(l * N)$. Based on the above, considering that in general $l << N$, the complexity for a single exectution of the *ClusterExtractor* is $O(N)$. *ClusterExtractor* will be executed once in case the network is connected, and few times, equal to the number of disconnected components, otherwise, thus, for sparse networks, the total complexity of *AutoWSCAN* is estimated at $O(N)$.

4.3 Synthetic benchmarks

Our initial hypothesis that WSCAN and AutoWSCAN are more suited for real world user interaction networks compared to SCAN needs experimental validation. Since, to our knowledge, there exist no real world weighted networks with ground truth communities, we utilize synthetic networks with planted partitioning of nodes in communities to evaluate the algorithms. In specific, we use the well-known LFR benchmark graphs [13], as they support weights and possess some important real world networks' features (node degree and community size heterogeneous distributions). Our benchmarking involves the application of WSCAN and SCAN on a series of LFR graphs generated with different parameters for several linearly increasing values of the parameter ε, while maintaining the same value for parameter μ. The accuracy of each run is evaluated by the well known Normalized Mutual Information (NMI) score [7], which quantifies the closeness between the identified and ground-truth communities in a scale of 0 to 1 (1 denotes identical assignment of nodes to communities). For each graph we record the best NMI score achieved and the corresponding ε value. To assess AutoWSCAN's performance, we apply it on the same graphs, and also compare it with a modified implementation for unweighted graphs, AutoSCAN. The latter follows exactly the same process as AutoWSCAN with the exception that is uses the classic (unweighted) measures of core reachability and reachability similarity.

SCAN-based approaches might characterize some nodes as outliers or hubs and not assign them to a community, as opposed to the LFR graphs which consider that each node belongs to at least one community. Since we are not aware of any weighted benchmark network with known community structure embedding also outliers and hubs, we adopt the LFR benchmark graphs and follow a workaround to extract NMI scores. Thus, upon the algorithms' execution, we assign i) outliers to the community with which they have at least one connection, and ii) hubs to the community towards which they are most strongly connected based on the (weighted) structural similarity or (weighted) reachability score for (W)SCAN and Auto(W)SCAN, respectively.

After obtaining the NMI scores for all approaches, we seek to reason their comparative performance by examining the benchmark graphs' structural properties. To this end, we employ two metrics: the global clustering coefficient and weighted clustering coefficient. The global clustering coefficient, CC, expresses the density of triplets of nodes in a network, where a *triplet* comprises three nodes connected by two (*open triplet*) or three edges (*closed triplet*). It is defined as 3 times the number of closed triplets (for each pair of the triangle's edges) over the total number of triplets at the network, and it ranges from 1 for a fully connected network to 0 for random networks with sufficiently large size. A similar idea is followed by the global weighted clustering coefficient, wCC, in weighted networks [21]. By assigning a value to each triplet, wCC is defined as the sum of all closed triplets' values over the sum of all triplets' values. Four methods have been proposed for the calculation of a triplet's value: the arithmetic mean, geometric mean, maximum, and minimum of the corresponding two edges' weights. Here, we employ the geometric mean method since it is considered the most appropriate for alleviating sensitivity to extreme weights. The definition of wCC implies that for a random distribution of weights in the network, wCC equals to CC. Here, for each network we calculate the ratio of wCC to CC and observe the performance of the algorithms when this ratio is greater or lower than 1.

4.4 Benchmark results

The proposed approaches, WSCAN and AutoWSCAN, are compared with their unweighted counterparts, SCAN and AutoSCAN, in terms of performance on the LFR benchmark framework. We aim to determine the validity of the proposed methods and their suitability for graphs with real world features. Since disregarding the variability of the intensity of interactions in real world networks is a common approach, here we try to identify how it affects performance and when it can be safely followed.

We evaluate the algorithms on four weighted LFR graphs, whose complexity is governed by the network *topological mixing* (μ_t) and *weighted mixing* (μ_w) parameters [13]. Since μ_t is the ratio of the number of a node's external neighbors to the node's total degree, its increasing values indicate mixed and difficult to separate communities. μ_w has a similar effect, since it is the ratio of the sum of the weights of the edges between a node and its neighbors in different communities to the sum of the all nodes' incident edges. Table 3 outlines the parameter combination for each benchmark graph. Benchmarks 1 and 3 refer to graphs with smaller communities (10-50 nodes per community) compared to Benchmarks 2 and 4 (20-100 nodes nodes per community). Also, graphs of Benchmarks 1 and 2 (with $\mu_t = 0.5$) have a more apparent community structure compared to Benchmarks 3 and 4 (with $\mu_t = 0.8$). Since we are interested in how weights affect the community detection results, we run SCAN, AutoSCAN, WSCAN and AutoWSCAN for varying values of μ_w.

Figure 2 depicts NMI scores for all runs on the four benchmark graphs (with $\mu = 4$). As expected, (Auto)SCAN performs invariably with respect to μ_t for all benchmarks, since it is not affected by changes at the edges' weights. (Auto)WSCAN's performance is satisfactory for the NMI score, as it starts to decay at $\mu_w \approx 0.5$. Lower NMI values are expected for high μ_w values, since then, the algorithms characterize more nodes as outliers/hubs and assign them to communities based on the workaround described in Section 4.3. For Benchmarks 1 and 2 the weighted algorithms perform better than (Auto)SCAN for $0.1 \leq \mu_t \leq 0.4$, while the corresponding set of graphs exhibit $wCC/CC > 1$ [9]. For $\mu_t \geq 0.5$ unweighted graphs maintain a good performance for Benchmark 1, whereas they perform poorly for all graphs of Benchmark 2 (with bigger communities and $CC < 0.1$). On the contrary, larger community sizes do not largely affect (Auto)WSCAN's performance, since NMI scores for Benchmarks 1 and 2, as well as for Benchmarks 3 and 4 are similar.

NMI scores from Benchmarks 3 and 4 indicate that the weighted algorithms perform better for $\mu_t = 0.8$, rather than for $\mu_t = 0.5$ (Benchmarks 1 and 2). This may seem contradictory, though as explained in [13], when $\mu_t < \mu_w$, inter-communities edges carry on average more weight rather than when $\mu_t > \mu_w$. This is inconsistent with most community detection algorithms' hypothesis that intra-community nodes are connected with highly-weighted edges. For all graphs of Benchmarks 3 and 4 the unweighted algorithms fail to detect the community structure. An important observation is that in these graphs $wCC/CC > 1$ (except on $\mu_t = 0.8$, where $wCC/CC = 1$). Results indicate that the decision of whether to apply (Auto)SCAN or (Auto)WSCAN on a given network could be based on the ratio wCC/CC, selecting the first when it is < 1, or the second otherwise. In all cases, automatic algorithms follow closely the best performance of their unweighted counterparts. This is a significant outcome given the temporal cost induced by the search of the (ε) parameter space in (W)SCAN. In our experiments, while the selected value for SCAN is always ~0.2, for WSCAN it increases for rising value of μ_w over all graphs with no common pattern. The selected ε value for all runs where WSCAN performs satisfactorily ($NMI > 0.5$) ranges from 0.04 to 0.28, it thus seems difficult to estimate it in advance. AutoWSCAN emerges as a good alternative to WSCAN, as it is independent of ε and performs similarly to WSCAN under the parameter setting leading to the best results.

Table 3 Synthetic Benchmark Graph Specification

	n	k	k_{max}	min_c	max_c	μ_t
Benchmark 1	5000	20	50	10	50	0.5
Benchmark 2	5000	20	50	20	100	0.5
Benchmark 3	5000	20	50	10	50	0.8
Benchmark 4	5000	20	50	20	100	0.8

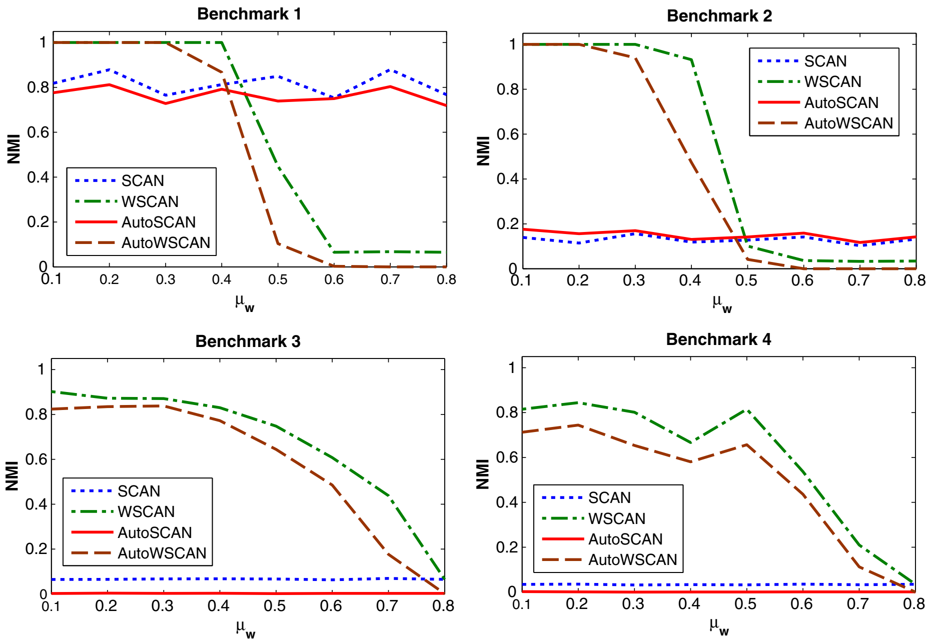


Figure 2 NMI scores for the algorithms' benchmarks with varying value of μ_w

5 Experiments and results

This section, presents the results of the proposed *global event*, *intra-event* and *cross-event* analysis derived by the framework's application in two real world case studies, EUROGROUP and TEDX. Experimentation involves networks from Twitter user interactions, namely *mentions*, *replies*, and *retweets*, generated from data collected via the Twitter Streaming API[4] using event-related topic keywords. Our selected case studies are described below and their characteristics are summarized in Table 4.

Recurring event case study The EUROGROUP case study refers to the official Eurogroup meetings, which have attracted major interest due to the recent financial crisis and Eurogroup's role in important decision taking. Covering 8 meetings from 13/06/12 to 30/11/12, it acts as an exemplary case study of a series of events held at different time instances, having the same participants with a common generic context (i.e. the eurozone's monetary issues), but different focus (depending on the agenda). Its impact is global since its decisions primarily affect the eurozones countries, but are reflected at a second level to the worldwide economy.

Modular event case study The TEDX case study is about a TEDx event that took place in Athens, Greece from 23/11/2012 to 24/11/2012. It concerns presentations, videos, and performances promoting creative or innovative ideas and projects. TEDX is used as an example of a local event with limited users, which is modular, since it takes place at a given

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

Table 4 Event-relevant features for the EUROGOUP and TEDX case studies

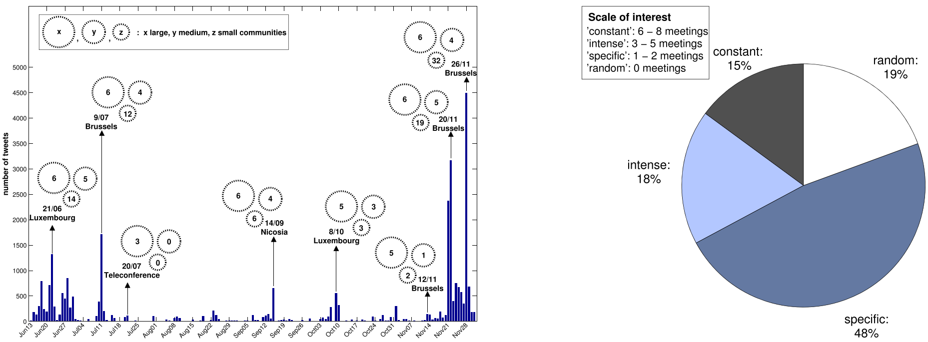
Feature	EUROGOUP	TEDX
Topic	focused	of wider scope
Impact scope	global	local
Intended audience	universal	closed community
Instance duration	1 day	2 hours
Instance granularity	sparser, aperiodic	continuous

time duration, but covers different topics in its sessions depending on presenters' expertise. Its monitoring duration spans from 21/11/2012 to 26/11/2012 covering the pre-event and after-event "talk" apart from the 2 main days.

5.1 Global analysis on the EUROGROUP case-study

First, we apply global event analysis on the EUROGROUP dataset. The total time duration of the dataset is 227 days and it comprises: 29529 tweets, 10305 interactions and 3015 different users. Regarding the interactions' type, retweets span more than 50 % of all interactions, thus they affect considerably the networks' shape (star-like forms). Statistical features such as tweet frequencies, depicted in Figure 3a, can be used to obtain some initial insights for an event's popularity in Twitter (e.g. more intense activity towards late November). Here, we are mostly interested in the users' clustering around such periods claiming that communities' emergent features reveal finer aspects of events. An extended version of this analysis can be found in [10].

To perform global event analysis, static community detection is needed, thus, we first normalize all weights and calculate wCC and CC for the user interaction network, resulting at a ratio of $wCC/CC = 1.22$. wCC is larger than CC implying that the intensities of user interactions are not random in this network, but play indeed an important role in communities' formation. Therefore, based on the observations of Section 4.4 we opt to apply AutoWSCAN for the detection of communities. AutoWSCAN reveals 67 communities



(a) (b)

Figure 3 EUROGROUP meetings, tweets, and communities: **a** depicts the daily number of tweets and is annotated by the meetings' dates and locations. The number of active communities per meeting is depicted above its corresponding day; **b** shows a distribution of the communities in a scale of users' interest based on their members' activity on the events' dates which we further analyze on the per community features of: size, topic diversity, and time span. To estimate the communities' topic diversity we apply LDA as described in Section 3.4. Since LDA requires specifying the number of topics to be detected, we empirically set this parameter to 100. Each document in LDA is a mixture of various topics with different probabilities. Here, due to the small length of tweets' text, a tweet is most likely to belong to a single topic, thus we assign it to the most probable one, and then calculate each community's topic diversity. The communities' *time span* is calculated at a daily granularity.

To understand each EUROGROUP meeting's impact, we associate them with the discovered communities and their features. We assume that each community expresses interest in an event, thus it is *active* on it, given that interactions between its members are observed on the current/previous/next day of the event. The number of active communities for each meeting are depicted in Figure 3a. To qualitatively characterize active communities, we further classify them as *small* (< 50 members), *medium* ($50 \leq$ members < 200), and *large* (≥ 200 members), and present their distribution for each event in the same figure. Since in total 6 large communities have been detected, we can observe that they are all active in 5 out of 8 events, which are also the events with the most tweets on the day they took place. Examination of the most popular events with respect to the tweets' number (20/11 and 26/11 in Brussels), reveals that although the latter has attracted the most tweets, the earlier has more medium active communities. The meeting of 20/11 corresponds to the failure of European leaders to *reach an understanding of how to restructure Greece's aid package, thus delaying the next aid tranche*, whereas this of 26/11 to the *IMF's and eurozone's €40 billion debt-reduction agreement for Greece*. [5] Although apparently more buzz was generated on the day of the later event, it seems that the previous, a long critical meeting building up tension and failing to reach a result, has attracted the interest of more large and medium communities combined. The later event, on the contrary, has been of interest to more small communities, probably focused on its decision. By comparing the summer meetings of 21/6 and 9/7, we can observe that although the first has attracted less tweets than the second, it is related to more communities which are active. June meeting's target was *to discuss the latest developments in the eurozone, mainly in Greece, Spain, Portugal and Ireland*, whereas July's meeting aimed at *discussing EU/IMF's rescue programs for Spain, Greece and Cyprus*[6]. More topics seem to be involved in the first event which may, up to a point, explain interest's dispersion in more communities. Some communities active on June's meeting might also be interested in a related topic: the announcement of the successful formation of a new government in Greece (after a critical long election period associated with the question of Greece's continued eurozone membership), which took place a day before the event. Communities are also characterized in terms of their interest in "Eurogroup" based on the number of meetings on which they are active. We assess interest expressed within a community in the following scale: *constant*, *intense*, *specific*, *random*, based on whether the community is active on 6-8, 3-5, 1-3, or 0 meetings, respectively. As depicted in Figure 3b, most communities appear to have *specific* interest on few meetings, though, a considerable percentage of them are indeed *focused* on the topic (with intense or constant interest).

[5] http://blogs.cfainstitute.org/investor/2011/11/21/european-sovereign-debt-crisis-overview-analysis-and-timeline-of-major-events/

[6] http://www.consilium.europa.eu/

To identify the most popular topics within tweets, we resort to the following approach. We form 3 orderings of topics by ranking each topic based on: A) the number of tweets that express it over all communities, B) the number of communities that are related to at least one tweet that expresses it, C) the number of communities that are *strongly-related* to it. For ordering C, we assign each community to a single topic, i.e. the one expressed in most of its members' interactions, and then we rank each topic by the number of communities assigned to it. A set of 12 unique topics is generated by taking the top-5 from each ordering. We define 3 topic features: *General Intensity* (GI), *Inter-Community Popularity* (ICP), and *Inter-Community Intensity* (ICI), which characterize topics that rank high (here, in the top-5) in ordering A, B, and C, respectively. In our set, there exist: 3 GI topics (which have the most intense user interest overall), 3 ICP topics (which reach out to the most communities), and 3 ICI topics (which play a major role in the most communities). There also exist 3 topics that combine two features, GI & ICP (attracting intense general interest while also being diffused in several communities), GI & ICI (attracting intense interest while also being major in several communities), and ICP & ICI (spanning several dedicated communities). Figure 4 depicts summaries of all 12 topics, where the central hexagons correspond to the GI, ICP, and ICI features, whereas the hexagons adjacent to two central ones represent the corresponding intersection. Topics are also divided based on their terms' language in

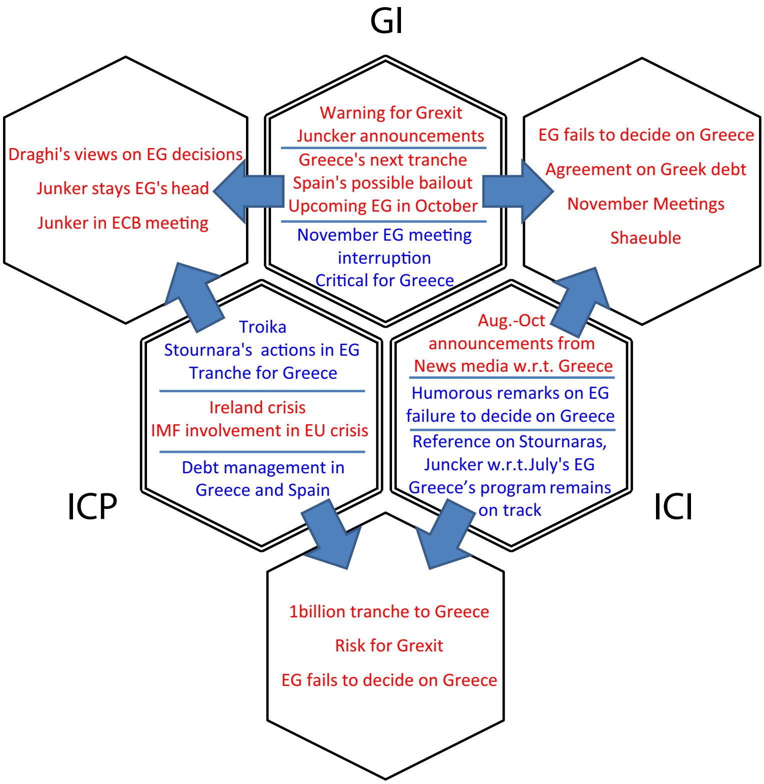


Figure 4 Classification of the most significant topics based on the intensity and diffusion of interest. *Horizontal lines* separate different topics, while topics in *red/blue* correspond to the English/Greek language. Greek topics have been translated in English. (*Best viewed in color*)

Table 5 Dataset features

Theme	Days	Tweets	Links	Users	Time frames	tu	$length(tf)$
EUROGROUP RT+	227	29529	10305	3015	167	1day	4
EUROGROUP RT-	227	29529	2755	997	167	1day	4
TEDX RT+	6	15247	9472	1815	69	2hours	1

English and Greek (the ones represented in the set). It can be easily observed that all topics that combine two features (thus more significant), are in English, indicating their impact on more users and communities.

5.2 Intra-event and inter-event analysis on the EUROGROUP and TEDX case-study

Next, we apply evolving community detection on both the EUROGROUP and TEDX datasets aiming to uncover more fine-grained users' interest fluctuations on the corresponding events via the proposed intra-event and inter-event analysis. Table 5 summarizes the two datasets based on their: time duration, total number of tweets for this period, number of interactions among users in the collected tweets, and the number of users. The table also presents the total number of time-frames analyzed for each dataset, along with the $length(tf)$ and tu used for the data's temporal analysis.

EUROGROUP dataset is analyzed with a 1-day step (tu) to a tf of 4 days, whereas the TEDX dataset is analyzed in 2-hour time-frames with no memory over previous ones (thus, the tu is also 2 hours). This setup was selected for the TEDX dataset in order to both approximately match each tf within the two days of the events with one of the event's sessions, and also avoid mixing tweets generated during different sessions. Despite its short duration, the percentage of interactions in TEDX tweets (~60 %) is almost twice the percentage of interactions in the EUROGROUP datasets. Figure 5 provides an overview of the TEDX dataset, with two easily-observable peaks in the number of tweets corresponding to the two days of the event, while some hours seem to be attract considerably more user interest than others.

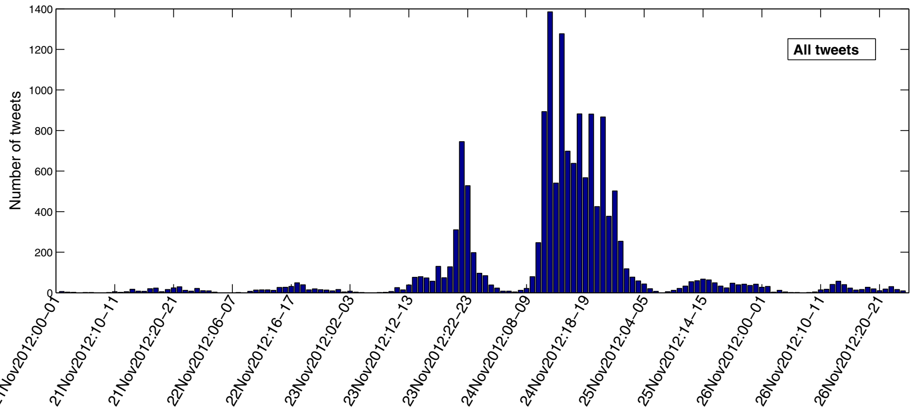


Figure 5 Total number of tweets per day in the TEDX dataset

Mentions and replies can be viewed as a more typical form of reference and communication between two users. Retweets, though, contribute not only to information propagation, but also to content validation and engagement in a broader conversation [5]. To this end, the retweets' role in the formation and evolution of user communities around events is studied by having two sets: EUROGROUP RT+ with all types of the above interactions, and EUROGROUP RT- with only mentions and replies.

Our framework identified the existing community chains in the dataset, and calculated their life span. The community chains' distribution based on their life span, depicted in Figure 6, indicates that few communities live for more than 4 time-frames for both EUROGROUP RT+ and RT- datasets, while the inclusion of retweets in general prolongs the communities' life span.

The EUROGROUP case-study Figure 7 illustrates the community detection results (number and mean size of communities) in all time-frames for both EUROGROUP RT+ and RT- datasets. Resulting diagrams are annotated with the eight Eurogroup meetings during the observation period. Figures 7a and 7c show e.g. that the last two Brussels meetings (end of November) have the largest dispersion in communities, while they are comparable to the Luxembourg meeting (June) with respect to their communities' strength. EUROGROUP RT- exhibits lower values on average for these two features, thus some events emerge as more *retweet-driven* than others. Such are the November Brussels meetings, where the communities' strength is significantly lowered by the retweets' removal (Figure 7d).

To characterize events, we apply our dual *intra-event* and *cross-event* analysis approach. We first perform intra-event analysis for the three EUROGROUP meetings of the biggest impact to community formation and tweet frequency, namely meetings of: Luxembourg (21/06), Brussels (9/07) and Brussels (20/11). The analysis' results are visualized for each event in Figure 8 via the so called *EventWheels*, which enable the joint presentation of the evolution of two networks based on several quantitative features corresponding to the same time-frames. Here, the EventWheels' left and right hemisphere correspond to EUROGROUP RT+ and RT- networks, respectively.

EventWheels comprise concentric circles representing the different time-frames relevant to an event instance, with the outermost and innermost circles representing the latest and earliest tf, respectively. The circles' number may vary depending on the event's actual

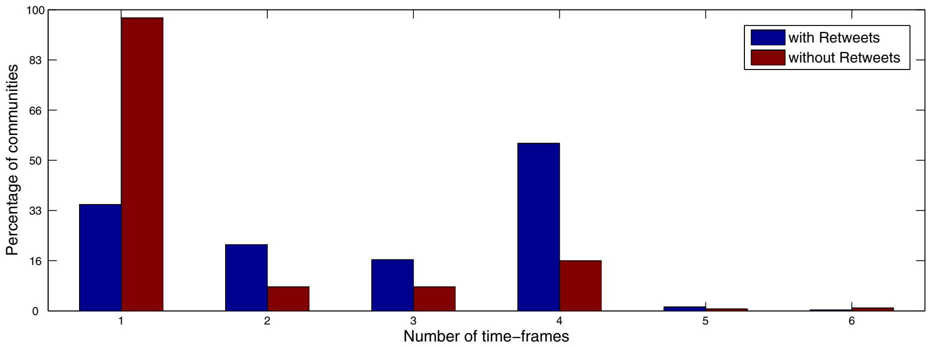


Figure 6 Distribution of communities in the EUROGROUP datasets based on their life span

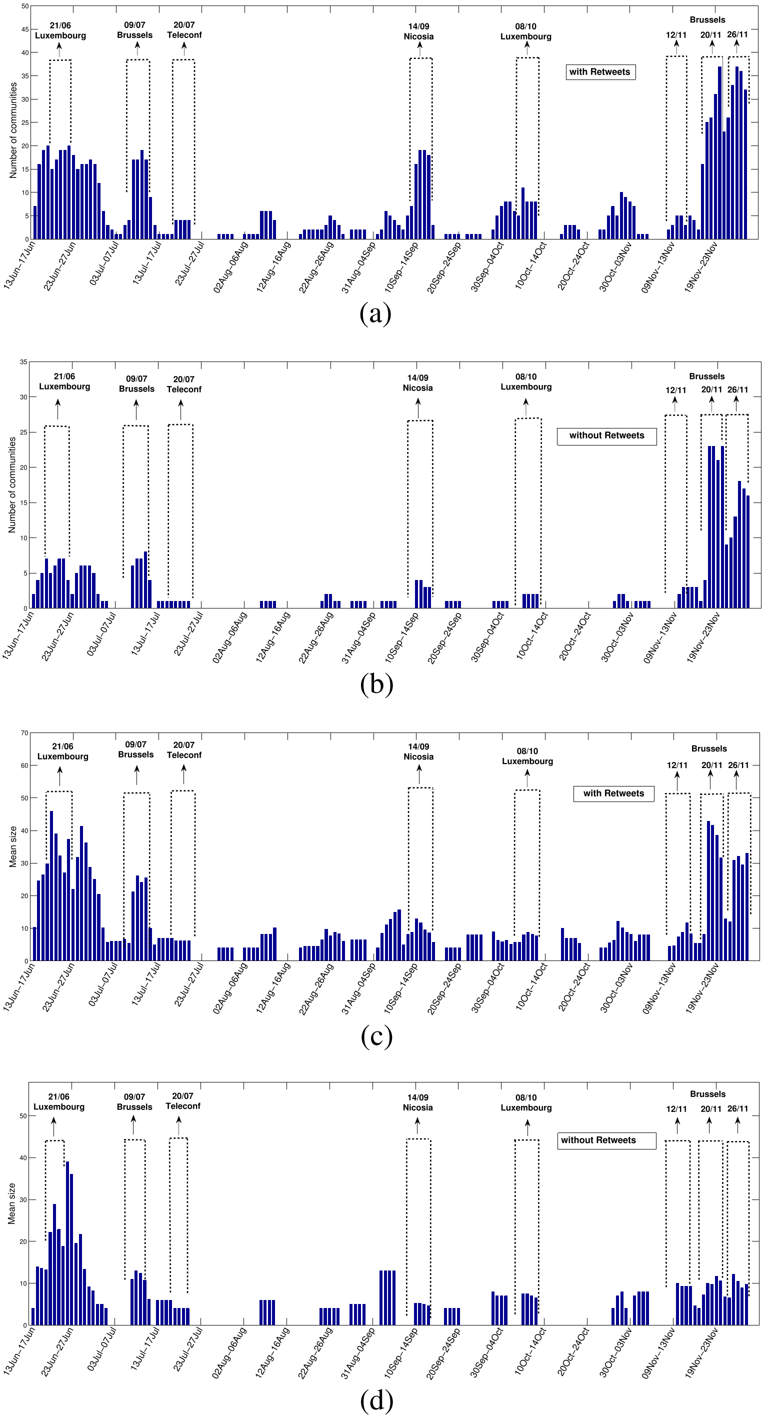


Figure 7 Community detection quantitative results for EUROGROUP: **a** and **b** depict the number of detected communities per tf, while **c** and **d** the mean community size, for R+ and R- datasets, respectively

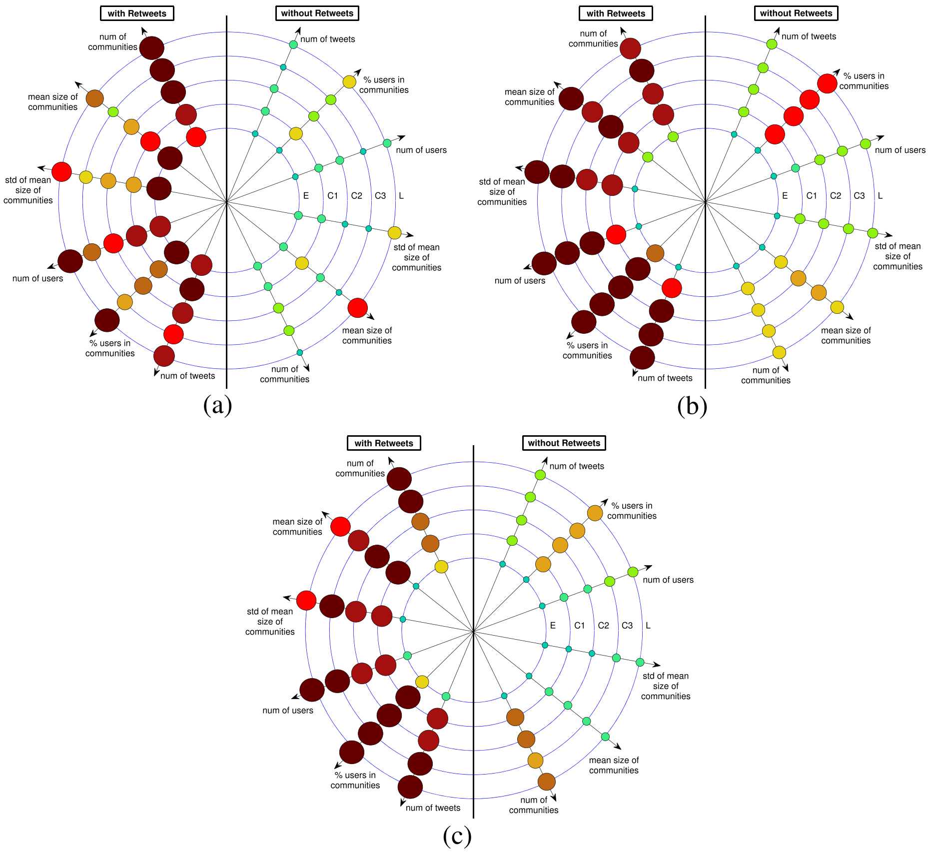


Figure 8 EventWheels for selected meetings in the EUROGROUP datasets which took place: **a** in Luxembourg on 21/06, **b** in Brussels on 9/07, **c** in Brussels on 20/11. *E* wheel represents the *early event* tf; *C1*, *C2*, *C3* wheels represent the *central event* time-frames; and *L* wheel represents the *late event* tf

duration, the $length(tf)$, and tu, while circles are categorized in: Early-event (E), Central-event (C), and Late-event (L), based on the time-frames' sequence and the distance of their center from the instance's central time point. Each EventWheel is intersected by a set of axes: one for each measured feature. Colored circles lie at these intersections, whose radius represents the relative value of the axis feature in the given circle's tf with respect to the other values in the axis of this event. EUROGOUP datasets have one E circle, three C circles, and one L circle (due to the selected $length(tf)$), as depicted in Figure 8.

Figure 8a for Luxembourg meeting shows strong user interest even from the event's beginning, as indicated by the communities' number, their mean size, and users involvement in the RT+ dataset. Interest on the event seems to weaken while it evolves, whereas in the L tf interest rejuvenates as indicated by the RT+ and especially the RT- dataset. In the RT+ dataset, users appear more dispersed at the event's end compared to its beginning, as the communities' number is governed by *expansion* forces while *shrinkage* forces are observed in the communities' mean size. The significant effect of retweets is also implied, as an opposite behavior is observed in the RT- dataset. Observations are inline with the actual event focus, discussed in Section 5.1, since it led to specific decisions for several countries of the agenda. Vivid interest at the event's end is justifiable as some decisions (for Greece: troika missions will be resumed, for Spain: formal assistance request is expected) left open unresolved issues.

Figures 8b and 8c depict the Brussels EventWheels with the 9/07 meeting having its most axes to be governed by *expansion* forces. Low values in the E circle indicate low interest and user networking at the beginning of the event, while soon boosted interest resulted in increased users' participation in communities and much more communities of varying sizes (but generally of a larger mean size). *Expansion* forces are observed in almost all axes, with the exception of the mean size of communities axis for the 20/11 event. While at their beginning, both meetings seem to be at the same scale, the decrease of the mean size of communities and the concurrent increase of both their number and the number of users towards the end of the 20/11 event (L circle), indicates that in this event users are much more dispersed compared to the 9/07 event. This large number of small communities indicates unresolved issues at the end of the 20/11 event, while on the contrary in the 9/07 event users seem to reach faster to some kind of consensus. These results are inline with the 9/07 event's positive conclusion of reaching political understanding on the Spain's recapitalization and financial institutions restructure program. Our results also capture the 20/11 event momentum of Eurogroup's failure to reach consensus, discussed in Section 5.1.

Table 6 depicts the EUROGROUP cross-event analysis with results indicating that, on average, in relatively small events (with respect to the number of tweets), community users are not well-connected. Life spans are comparable in all events, however it seems that smaller events have a slightly prolonged life span. By comparing the Luxembourg meeting of June and the Brussels meeting of 20/11, we can observe that in the first one the percentage of clustered users versus outliers is higher, however the number of communities is considerably smaller. This may indicate that in the Luxembourg event most users are connected and gathered in fewer and on average stronger communities, whereas in the Brussels event there is more dispersion, having a higher number of outliers and several communities of smaller scale.

To zoom-in a specific community chain we indicatively pick Brussels 20/11 EUROGROUP RT+ dataset in C1 tf and follow its evolution across time in Figure 9 (link's color indicates type, with 'blue' for retweets, 'red' for mentions, and 'yellow' for replies). Here, mentions are shown to be stronger than replies and retweets regarding their weight, and retweets create star-like structures in communities, as anticipated. Figures 9a, 9b and 9c show the community's rise from C1 to C3, and its split in smaller communities in L,

Table 6 Cross-event features for the EUROGROUP (RT+/RT-) datasets

Event	21/06 Luxembourg	09/07 Brussels	20/07 Telecon.	14/09 Nicosia	08/10 Luxembourg	12/11 Brussels	20/11 Brussels	26/11 Brussels
Time period	17-25Jun	5-13Jul	16-24Jul	9-17Sep	4-12Oct	8-16Nov	16-24Nov	22-30Nov
mean life span	3/3	**4**/3	**4**/ **4**	**4**/ **4**	**4**/ **4**	**4**/3	3/ **4**	3/ **4**
tweets	2613/718	789/227	49/23	335/71	249/79	161/68	**4431/1037**	2805/659
communities	46/18	27/3	4/1	26/4	14/2	7/4	62/ **34**	**77**/29
users in com	1372	471	479	230	95	44	1294	**1395**
users in com. (%)	**77**	62	60	63	30	35	70	66
retweets (%)	73	71	53	**79**	68	58	76	77

Bold entries indicate the highest value per feature

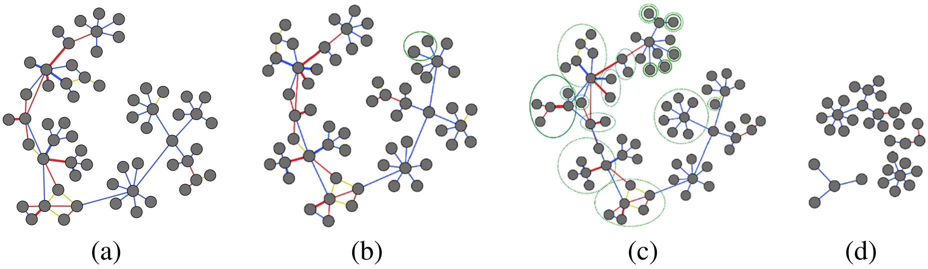


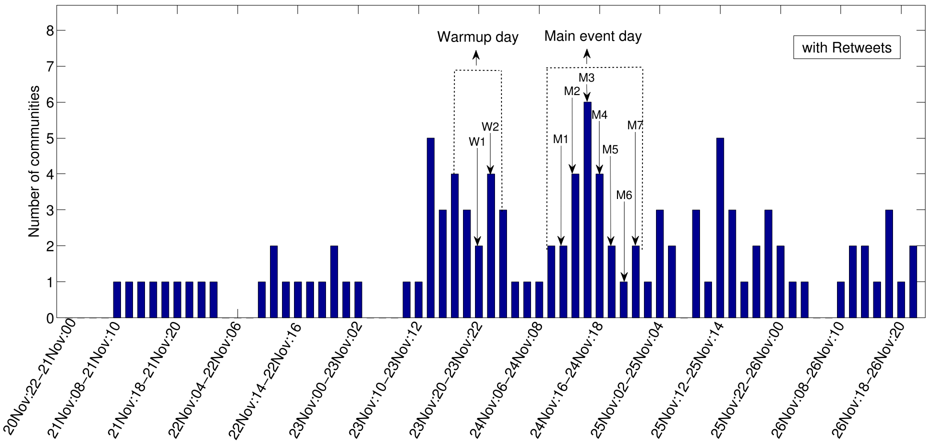
Figure 9 Evolution of a EUROGROUP RT+ community within C1,C2, C3, and L time-frames. **a** depicts the community's *birth* in C1, **b** its evolution into a slightly larger community in C2, **c** a larger size increase in C3, and **d** its split into a number of small communities in L. *Green straight/dashed circles* indicate the enclosed nodes' appearance/disappearance in the current/next tf. (*Best viewed in color*)

indicating that this community chain was created in the event's start, and gradually, as the event progressed, more users with similar interest joined it, leading to its dissolve near its ending. Table 7 shows some popular tweets exchanged between users of two snapshots of this community chain taken in C1 and C3, comprising both news-broadcasts and opinions. Tweets in C1 show the dissatisfaction of this community's users and their low expectations of the event, while tweets in C3, when the event's outcome has almost been stabilized, show aspects of official Eurogroup members' statements and news media articles on the event.

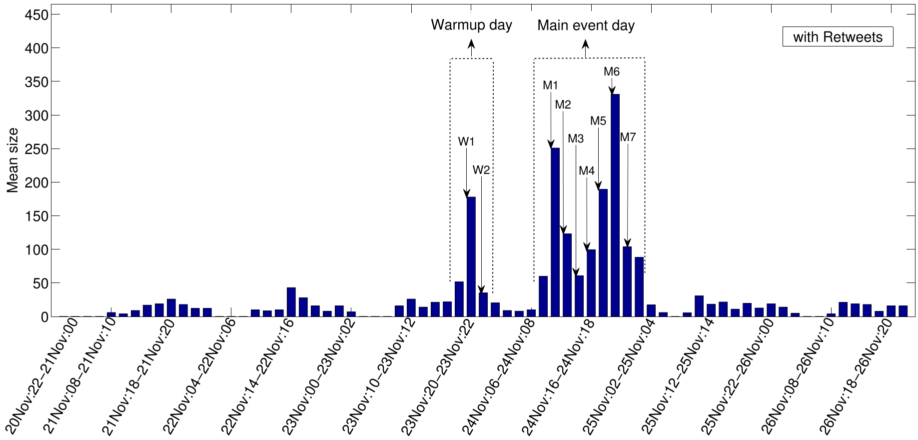
The TEDX case-study Community detection results for the TEDX dataset, in terms of the number and mean size of communities per tf, are depicted in Figures 10a) and 10b, respectively. The event spans the warmup (23/11) and main day (24/11), and is analyzed in 2-hours sessions (corresponding to the actual event sessions), which we handle similarly to event instances. Identifiers W1-W2 are used to reference the warmup day's time-frames (18:00-22:00), and identifiers M1-M7 are used for the main day's time-frames (08:00-22:00). Diagrams in Figure 10 are annotated with these sessions and assist in quantifying their effect on users' interactions. These diagrams combined with the cross-event features of Table 8 indicate that in the warmup day, users' interactions in Twitter are more intense during the first session (W1), whereas in W2, a small number of users is distributed in four communities. The main day attracted much more interactions with vivid users networking even from M1 (characterized by few communities of relatively large mean size), indicating

Table 7 Popular tweets within the communities depicted in Figure 9a (C1 tf) and Figure 9c (C3 tf)

C1	Anyone still want to take my bet? That Eurogroup will approve Greek bailout cash but there's no way it will pass national parliaments
	I have #Eurogroup fatigue, my poor journalist friends have Eurogroup fatigue heck even the Eurogroup is bored with itself
	Another Eurogroup "thriller" about Greek debt
C3	Dear #Eurogroup Serious talk of #Greek bond buy back = increase in mkt prices. OK?
	Talk of #Grexit = fall in mkt prices
	Another Eurogroup on Greece needed after Bundestag vote, before money can be disbursed, says German govt spokeswoman.
	We had to destroy Greece to save it #EU #IMF



(a)



(b)

Figure 10 Community detection quantitative results for the TEDX dataset depicting: **a** the number of detected communities per tf; **b** the mean community size per tf

low dispersion and high communities' strength, whereas M3, a session in the middle of the event's main day, is characterized by high dispersion and small communities, possibly as discussions may also involve presentations of previous sessions.

Table 8 Cross-event features for the TEDX dataset

Event	W1	W2	M1	M2	M3	M4	M5	M6	M7
tweets	834	234	**1410**	1177	766	833	884	822	504
communities	3	4	2	4	**6**	4	2	1	2
users in com.	712	282	**1004**	984	730	796	758	662	416
users in com. (%)	93	84	**94**	92	89	90	89	93	89
retweets (%)	69	65	69	65	69	69	69	**70**	**70**

Table 9 Most frequent terms in four communities of the M3 event session of the TEDX dataset

com. index	most frequent terms per community
1	sam conniff, marykatrantzou, walzer, tgeorgakopoulos, papadimitriou, failure, bookshop, offenders
2	craig, walzer, craig walzer, emeaportal, stakon, alex_walex, santorini, bookstore, no more ebooks
3	adiasistos, 45 euro, ticket, tweet all time, rotten innovation
4	news247gr, greece eats its children, sam conniff

To gain some insight in the thematic focus of communities discovered within a given tf we examine tweets in four communities of the M2 session. Upon cleaning the tweets, we perform a most frequent n-gram analysis ($n \in 1, 4$), and present the most frequent terms/ phrases within each community (Table 9). M3 session had four speakers: Sam Conniff (Social Entrepreneur), Mary Katrantzou (Fashion Designer), Craig Walzer (Founder of Atlantis Bookstore), and Andreas Mershin (Research Scientist). Community 1 is of mixed theme, but is mainly dominated by references to the presentation of Conniff, and especially his proposal about the rehabilitation of ex-offenders into society with the aid of technology. Other topics involve presentations of Katrantzou and Walzer, Paul Papadimitriou, a presenter of the previous session, and the event's live streaming. Community 2 is more focused in topic, since it revolves mainly around Walzer and his inspired bookshop in Santorini, Greece. On the other hand, community 3 is rather driven by opinions regarding the event itself, comprising mostly users who express negative opinions about the event organization, complaining about the ticket's price and criticizing its innovation focus. Finally, community 4 is a small group of users of limited topic, mainly discussing the phrase *greece eats its own children* included in Conniff's presentation.

The comparison of results from EUROGROUP and TEDX indicates that in TEDX there is a lower outliers percentage compared to EUROGROUP. Also, it generally has a much smaller number of communities, while the total number of users in communities is comparable for both cases. This is reasonable since a TEDX tf is constrained to 2 hours, thus users' interest concentrates around few topics, whereas a EUROGROUP tf covers 4 days, thus a larger sub-topic variety.

6 Conclusions

This work's contribution was to: i) present a generic framework for community tracking in real world interaction networks, instantiated with a suitable algorithm, ii) identify community as well as community chain features which can be leveraged for revealing events impact on social media users and their interactions, iii) propose approaches for global-, intra-, and cross- event analysis and demonstrate their potential on two exemplary event-related case studies. Role of weights in community detection approaches is studied based on structural similarity and on automatic parameter selection. Our proposed community detection approach leverages network's structural properties and interactions intensities and it is validated over a series of synthetic networks. The three proposed event analysis approaches have revealed different aspects of the underlying events, and exhibit different merits. In specific, global event analysis provides a generic overview of relevant social media users' activity, while the other two approaches provide a zoom-in analysis, at different granularities, by identifying communities in a time period which is synchronized with a given event instance. Revealing reciprocities among people communities and real word events via social media analysis embeds significant challenges with several promising future research directions. Next research work is foreseen in studying and assessing the effect of the tf and tu parameters, and extending our approach further under a different scenario, where the event instances and their duration are not known in advance, but should be rather derived based on the monitored activity in social media. Another interesting future extension is to proceed to qualitative assessment of the identified communities by a methodology which will capture their polarity with respect to the underlying events and will identify users roles within and across them.

Acknowledgements This research was supported by Qualia, an on-line media monitoring and business intelligence company, in Athens, and the Research Committee of the Aristotle University of Thessaloniki.

References

1. Ankerst, M., et al.: Optics: ordering points to identify the clustering structure. SIGMOD, pp. 49–60 (1999)
2. Asur, S., et al. ACM Trans. Knowl. Discov. Data **3**(4), 16:1-16:36 (2009)
3. Blei, D.M., et al.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**(4-5), 993–1022 (2003)
4. Bortner, D., Han, J.: Progressive clustering of networks using structure-connected order of traversal. ICDE, pp. 653–656 (2010)
5. Boyd, D., et al.: Tweet, tweet, retweet: Conversational aspects of retweeting on Twitter. HICSS, pp. 1–10 (2010)
6. Choudhury, M.D., et al.: Unfolding the event landscape on twitter: classification and exploration of user categories. CSCW, pp. 241–244 (2012)
7. Danon, L.A., Díaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. J. Stat. Mech.: Theor. Exp. **2005**(9), P09008 (2005). http://stacks.iop.org/1742-5468/2005/i=09/a=P09008
8. Ester, M., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. KDD, pp. 226–231 (1996)
9. Giatsoglou, M., et al.: Community detection in social media by leveraging interactions and intensities. WISE **2**, 57–72 (2013)
10. Giatsoglou, M., Vakali, A.: Capturing social data evolution using graph clustering. IEEE Internet Comput. **17**(1), 74–79 (2013)
11. Jurgens, D., Lu, T.-C.: Friends, enemies, and lovers: detecting communities in networks where relationships matter. WebScience (2012)
12. Kamath, K.Y., Caverle, J.: Transient crowd discovery on the real-time social web. WSDM, pp. 585–594 (2011)
13. Lancichinetti, A., Fortunato, S.: Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Phys. Rev. E **80**(1), 016118 (2009). doi:10.1103/PhysRevE.80.016118
14. Leskovec, J., et al.: Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. CoRR, abs/0810. 1355 (2008)
15. Leung, I.X., et al.: Towards real-time community detection in large networks. Phys. Rev. E **79**(066107) (2009)
16. Lin, C.X., et al.: Pet: a statistical model for popular events tracking in social communities. KDD, pp. 929–938 (2010)
17. Lin, Y.R., et al.: Metafac: community discovery via relational hypergraph factorization. KDD, pp. 527–536 (2009)
18. Marcus, A., et al.: Twitinfo: aggregating and visualizing microblogs for event exploration. In: CHI, pp. 227–236 (2011)
19. Morrison, D., et al.: Evolutionary clustering and analysis of user behaviour in online forums. ICWSM, pp. 519–522 (2012)

20. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004). doi:10.1103/PhysRevE.69.026113
21. Opsahl, T., Panzarasa, P.: Clustering in weighted networks. Soc. Networks **31**(2), 155–163 (2009)
22. Palla, G., et al.: Quantifying social group evolution. Nat. **446**, 664–667 (2007)
23. Papadopoulos, S., et al.: Community detection in social media. Data Min. Knowl. Disc. **24**, 515–554 (2012)
24. Quack, T., et al.: World-scale mining of objects and events from community photo collections. CIVR, pp. 47–56 (2008)
25. Sander, J., et al.: Automatic extraction of clusters from hierarchical clustering representations. PAKDD, pp. 75–87 (2003)
26. Shamma, D.A., et al.: Conversational shadows: Describing live media events using short messages. ICWSM (2010)
27. Sun, J., et al.: Graphscope: parameter-free mining of large time-evolving graphs. KDD, pp. 687–696 (2007)
28. Sun, H., et al.: gskeletonclu: Density-based network clustering via structure-connected tree division or agglomeration. In ICDM, pp. 481–490 (2010)
29. Takaffoli, M., et al.: Modec - modeling and detecting evolutions of communities. ICWSM, pp. 626–629 (2011)
30. Tang, L., et al.: Community detection via heterogeneous interaction analysis. Data Min. Knowl. Disc. **25**(1), 1–33 (2012)
31. Xu, X., et al.: Scan: a structural clustering algorithm for networks. KDD, pp. 824–833 (2007)
32. Zhao, Q., et al.: Temporal and information flow based event detection from social text streams. AAAI, pp. 1501–1506 (2007)