Analyzing Users' Activity in On-line Social Networks over Time through a Multi-Agent Framework

E. del Val, C. Martínez, and V. Botti

Universitat Politècnica de València
{edeval, cmartinez1, vbotti}@dsic.upv.es

Abstract. The number of people and organizations using on-line social networks as a new way of communication is continually increasing. Messages that users write in networks and their interactions with other users leave a digital trace that is recorded. In order to understand what is going on in these virtual environments, it is necessary systems that collect, process, and analyze the information generated. The majority of existing tools analyze information related to an on-line event once it has finished or in a specific point of time (i.e., without considering an in-depth analysis of the evolution of users' activity during the event). They focus on an analysis based on statistics about the quantity of information generated in an event. In this article, we present a multi-agent system (MAS) that automates the process of gathering data from users' activity in social networks and performs an in-depth analysis of the evolution of social behavior at different levels of granularity in on-line events based on network theory metrics. We evaluated its functionality analyzing users' activity in events on Twitter.

1 Introduction

The way people communicate with each other is changing [28]. On-line Social networks such as Flickr, Linkedin, Facebook and Twitter contain millions of users and are among the most popular sites on the web. Currently, users (i.e., individuals, companies, associations, media, etc.) share their thoughts, preferences, feelings, or political beliefs in on-line social networks. According to [14], over 50% of internet users aged between 16 and 54 used on-line social networking sites and, on-line social networks remained enterprises' favorite form of social media [15]. In these networks, each user's contribution or interaction with others leaves a digital trace. Therefore, there are vast amounts of data that can be used for research on human behavior [43].

Data generated by users in on-line social networks can be modelled as networks where nodes represent users and links represent social ties among them (i.e., friendship, interaction, location, professional, etc.). According to Huberman et. al [20], the hidden social network among users that interact with each other is the one that really matters when trying to rely on word of mouth to spread an idea, a belief, or a trend. However, in general, previous works that analyze on-line social networks focus on the analysis of friendship relationships [8, 40, 39, 21] instead of interactions among users. In the area of marketing, the analysis of on-line social networks has been applied to commercial usage, marketing and strategies of persuasion [27], recommender systems [19] and also as social sensor to predict outcomes [7] and to determine potential consumers [2]. It has also been used to determine the users' personality [26, 18], to detect the most influential users [35], or to understand how information flows [36]. Social network analysis has also been applied to social contexts such as terrorism [34] and cyberbullying [3].

Social networks can be considered dynamic processes [37, 42] where, as time passes, individuals join, leave, create or deactivate social ties thereby altering the structure of the network [24]. The analysis of the evolution of interaction networks in real-time is considered a valuable source of information about human social behavior [12]. Monitoring users' activity in real time and performing an analysis is however a complex task for humans, due to the well-known problem of information overload. The main problem is how to automate the process of monitoring user interactions with the intention of detecting behavior patterns in users and/or groups in a distributed and efficient way to deal with users' demand.

In order to deal with this problem, we propose the use of Multi-Agent Systems [22, 11]. Social networks can be well understood via a multi-agent perspective. MAS is an appropriate technology to deal with the distributed analysis of the evolution of users activity in on-line social networks. MAS are well suited to implement complex, distributed software systems. Agents as social computing entities are able to collaborate in the process of collecting, processing, and analyzing information in a distributed way. MAS can adapt to changes in the environment while still achieving overall system goals, dealing with the high dynamicity of Internet. Moreover, MAS provide a foundation to create an architecture that deals with the complexity reduction, flexibility, and scalability needed in the automatic analysis of users' activity in events in on-line social networks.

In this paper, we propose a MAS to automatize the process of gathering data from users' activity in on-line social networks and analyze the evolution of this activity in events at different levels of granularity based on network theory metrics. The proposed MAS is organized in virtual organizations where there is a set of roles (i.e., monitor, analyzer, manager, and user) that defines the services offered. The most relevant role is the analyzer that is characterized by offering services related to a configurable analysis of the collected information that is associated to an event. These services provide an analysis of what is happening before, during and after an event at global and individual level. All the information generated during the analysis is provided to the final user through a web interface. Agents that are part of the MAS operate asynchronously and concurrently, which results in computational efficiency. New requirements or functionalities can be addressed by adding new agents or by reconfiguring the existing ones.

The remainder of this article is organized as follows. Section 2 shows the most relevant works in the literature with regard to on-line social networks analysis. Section 3 presents the framework based on a MAS architecture and describes the framework dynamics. Section 4, presents a case study of several events in Twitter using the proposed MAS. Finally, Section 5 presents some concluding remarks.

2 Related Work

The majority of real-world on-line social networks evolves over time and generates an enormous amount of data that is temporally annotated. Currently, a limited number of works have analyzed in depth the evolution of users' activities in on-line social networks. There are previous works that analyze the evolution over time of some properties or metrics. For instance, Meeyoung et. al [9] consider friendship and favorite marking relationships in Flickr. The authors perform an analysis that includes network topology (i.e., path length, clustering coefficient, diameter, and degree distribution) at the end of a period of time. The metric that is analyzed over time is the number of fans per day associated to photos. Benevenuto et. al [4] consider Orkut, MySpace, Hi5, and LinkedIn social networks. Their analysis focuses on detailed click-stream data. They analyze metrics related to the users' sessions in social networks. The metric that is analyzed over time is the number of unique on-line users. Kumar et. al [25] presented a more complete evolutionary analysis of structural properties in *Flickr* and *Yahoo! 360* networks. However, the analysis is performed over friendship networks that does not provide a realistic view of relationships. Borge-Hoelthoefer et. al [6] studied the structural and dynamic patterns of a network made up of Twitter users during the 15M social phenomenon in Spain. They analyze the evolution of the giant component, the tweets received and sent, and the popularity. There are other properties such as in- and out-strength distributions and community structure that are analyzed in a specific point of time. There are other works that focus on understanding the users' interactions related to political events and analyze the evolution of some properties in specific points of time [7, 29, 33]. Borondo et. al [7] analyze topological properties of mention and retweet networks, the cumulative distribution for users' activity, and the percentage of retweets and mentions targeted to politicians and mass media official accounts at the end of the event. They only perform an analysis of the evolution of accumulated tweets over time. Lotan et. al [29] analyze Tunisian and Egyptian revolutions in Twitter. They analyze the distribution of actor types over the users that participate in the events, the participation of organization vs individual accounts, and information about the number of messages generated in both events. The analysis of the evolution of properties of the events as time passes is only focused on the number of tweets. Peña et. al [33] analyze a socio-political event in Spain called 15M and compare it with the 25S. The authors analyze the participation of users in the events, the distribution of types of messages, demographical characterization, and community detection once the event finished. The authors provide several figures that show the evolution of relationships established during each stage of the 15M.

Summarizing, the existing works analyze the majority of metrics associated to users' activity at the end of the event (see Table 1). They do not provide an analysis in real-time. In some of them, there is one specific property that is analyzed during a period of time, but in general, properties are analyzed over the whole event without considering their evolution. In the approaches presented by Benevenuto et. al, Kumar et. al, and Meeyoung et. al the authors focus on friendship relationships instead of interactions, that provide a less realistic view of the relationships. Our proposal tries to overcome these limitations providing a multi-agent framework that automatizes the information retrieval and a depth analysis of the evolution of metrics associated to an event in on-line social networks. The analysis can be performed in real-time and after the event and considers the interaction among users as relationships. The analysis that our proposal offers can be configurable and provide different levels of detail (i.e., network and individual level). The use of MAS technology facilitates the distribution an scalability of the data collection, processing, and analysis of users' activity considering users' demand.

Approach	Analysis over time?	Network Links	Where?	When?	What?	Configurable Analysis Framework
Meeyoung et. al	one property: of fans associated to photos (day)	friendship and favorite marking	Flickr	after the event	path length, clustering coefficient, diameter, degree distribution, fans per day.	✗
Benevenuto et. al	one property: on-line users (hour)	click-stream data	Orkut, MySpace, Hi5, and LinkedIn	after the event	metrics related to users' sessions	✗
Kumar et. al	✓ (week)	friendship	Flickr Yahoo! 360	after the event	singletons, giant component, middle region	✗
Borge-Hoelthoefer et. al	✓ giant component tweets received /sent popularity	interaction Events: 15M	Twitter	after the event	Network growth, community structure, information sources and sinks	✗
Borondo et. al	✓ accumulated tweets	interaction Spanish president election	Twitter	after the event	users activity, topological properties, messages targeted to politicians.	✗
Lotan et. al	only number of tweets over time	interaction	Twitter	after the event	actor types, participation of organization vs individuals, num. of messages	✗
Peña-Lopez et. al	only relationships over time	interaction	Twitter	after the event	type of messages, demographics, community detection	✗
Proposal	✓ (configurable)	interaction	Twitter	during and after	network level individual level	✓

Table 1. On-line Social Network Analysis Approaches.

3 MAS Framework

In this section, we present a MAS framework for the analysis of the evolution of social users' activity related to a specific event in on-line social networks. The following agents compose the MAS: *Manager*, *Monitor*, *Analyzer*, and *User*. The *User* agent is the intermediary between the MAS and the final user. The user interacts with this agent to indicate which event(s) he/she is interested in. The *User* agent receives this information and sends a message to the *Manager* to start tracking the event(s). The *Manager* is responsible for managing the events selected by users. The *Manager* takes into account how many users are interested in tracking the social behavior of users for each event and the size of the events (i.e., the number of messages generated by users) in order to create one or more *Monitor* agents. *Monitor* agents are responsible for collecting users' messages associated to an event. These messages are stored in a database. The *Analyzer*

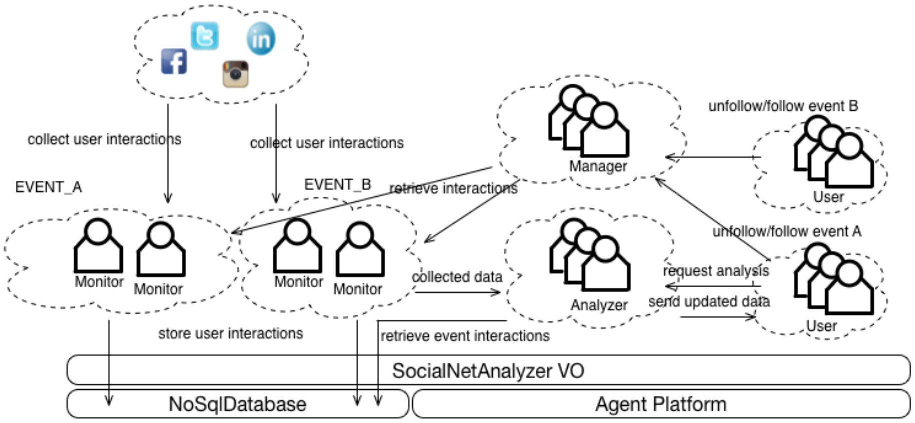


Fig. 1. MAS framework for the analysis of the evolution of users' activity during an event in on-line social networks.

agent performs an analysis on user demand. Once one or several *Analyzer* agents, depending on the size of the event (i.e., the users' activity), perform the analysis, all the data is sent to the *User* agent. The *User* agent is responsible for collecting all the results provided by *analyzer* agents and displaying the results of the users' activity analysis associated to the event to the user through a web interface.

In the following paragraphs, we describe with detail the main components of the framework (see Figure 1): *agent platform*, *data model*, and *agents*. The framework focuses on the analysis of users' activity during events in Twitter. However, it can be used in other on-line social networks.

Agent Platform. The framework consists on a set of agents that are developed using Magentix2 platform [1]. Magentix2 is a platform for open MAS that supports flexible interaction protocols and conversations, indirect communication and interactions among agent organizations [16]. Also, the platform permits the definition of a normative context, which can be easily adapted to the changes in the system requirements [41]. Nowadays, Magentix2 provides support at three levels: *Organizational level* that consists on technologies and techniques related to agent societies; *Interaction level* that supports flexible interaction protocols and conversations; and *Agent level* that offers different classes of agents such as *CAgent* (i.e., an agent that allows the automatic creation of simultaneous conversations) or *JasonAgent* (i.e., BDI agents that can participate in simultaneous conversations).

Data Model. In order to store, process, and query all the data about the activity of users during an event, we used a graph data model based on Neo4J database[2]. Data models in

[1] http://gti-ia.upv.es/sma/tools/magentix2/index.php
[2] http://neo4j.com/

Neo4J consider three components: *labels* that are used to group nodes into sets, *nodes* (i.e., entities), and *relationships* that connect nodes and organize them into arbitrary structures.

In our proposed model, we have considered the following nodes (entities):

- *hashtag*: represents the event that a user is interested in. We identified an attribute associated to the hashtag that is the name of the event.
- *message*: represents the message (i.e., text message, photo, video, etc.)that a user creates in a on-line social network. We identified a set of attributes associated to the message such as coordinates, temporal annotations, references to other users, number of likes/favourites, etc.
- *user*: represents an user that has participated through the publication of a message or has been mentioned in the content of a message in an on-line social network. We identified a set of attributes associated to the user that were extracted from the message such as temporal annotations, friends/followers, likes, etc. depending on the on-line social network.

The relationships that connect nodes are:

- *has_hashtag*, which connects a message with an event;
- *mentions*, which connects a message (i.e., text message, photo, video, etc.) with one or several users that are referenced in the message; and
- *creates*, which connects a user with a message.

Using these entities and relationships, we were able to efficiently store all the information generated by users about a certain event in on-line social networks. As as example, we show the use of this model for the case of storing the activity of users during an event in Twitter but it can extrapolated to other social networks where interactions could be modelled as messages among users about a topic or event. In Twitter, we considered the following type of messages:

- *Global messages* that are used when the update is meant for anyone that cares to read it. This information and its relationships are represented using the Neo4j model as follows:

  ```
  (u:USER)-[:CREATES]->(t:MESSAGE)
  (t:MESSAGE)-[:HAS_HASHTAG]->(e:EVENT)
  ```

- *Individual messages* that are those that involve another user. Individual messages in Twitter can be: retweets, mentions, and replies to users.
  - Retweets are messages that were previously posted by another user.
  - Mentions are messages that are used when a user aims to inform about an update to a specific person. Often, two or more users will have conversations by posting mentions to each other.
  - Replies to users are messages where a user mentions another user as a result of a previous message.

  Information about individual messages and their relationships are represented using the Neo4j model as follows:

  ```
  (u:USER)<-[:MENTIONS]-(t:MESSAGE)
  ```

Agents. The framework consists on a set of agents that play one of the following roles: *User*, *Manager*, *Monitor* and *Analyzer*. The functionality of each agent is implemented as a set of independent services that can be allocated in any host, even in different ones. All the agents of the system are part of the *SocialNetAnalyzer* Organizational Unit and extend the *BaseAgent* provided by Magentix2 platform. Agents are developed using Java. Depending on the organizational role that an agent plays and the on-line social network considered, it makes use of certain external functionalities.

- *Monitor* agent makes use of API's to facilitate the retrieval of the public data generated by users during an event. Data related to users activity in on-line social networks is stored in Neo4J. Neo4j is an open-source NoSQL graph database implemented in Java and Scala. Neo4j is a highly scalable native graph database that uses relationships as a core aspect of its data model and is able to store, process, and query connections efficiently.
- *Analyzer* agent makes use of Cypher[3]. Cypher is a declarative graph query language that allows for expressive and efficient querying and updating of the graph store. Cypher borrows its structure from SQLqueries. Clauses are chained together, and feeds intermediate result sets between each other. *Analyzer* agent uses Cypher to get information about users' interactions associated to an event during a period of time. As an example, we show an instance of a query in Cypher for the information retrieval in Twitter:

  ```
  MATCH (u:USER)-[:CREATES]->(t:MESSAGE)-[:HAS_HASHTAG]
  ->(e:EVENT{Hashtag:"myEvent"}),
  OPTIONAL MATCH (u2:USER)<-[:MENTION]-(t:MESSAGE),
  WHERE t.tweet_created_at>="time_start"
  AND t.tweet_created_at<="time_end",
  RETURN u.User_screen_name, t.tweet_id, u2.User_screen_name)
  ```

  Based on the information retrieved from Neo4j database, the *Analyzer* agent builds a temporal annotated interaction network. A user $\mathbf{A}$ becomes a node of the network when he/she participates by writing a global or individual message with the hashtag associated to the event or when another user $\mathbf{B}$ references him/her in an individual message inside the temporal interval selected. Each node of the network (user) has an associated label that represents the instant when he/she joined the network. Links of the network are established when a user writes an individual message to an existing or new user. Therefore, the network is directed (see Figure 2). It is important to note that we work with accumulated data (i.e., the network at time t also includes the nodes and links from a previous moment t' $<$ t). The analyzer agent creates the network using GEXF (Graph Exchange XML Format)[4], a language for describing complex networks structures, their associated data and dynamics.
  Once the *Analyzer* agent has built the network of user interactions associated to an event during a temporal interval, it starts the analysis of the structural properties at global and/or individual level. To carry out with this task, the agent uses Gephi

[3] http://neo4j.com/developer/cypher-query-language/
[4] https://gephi.org/gexf/format/

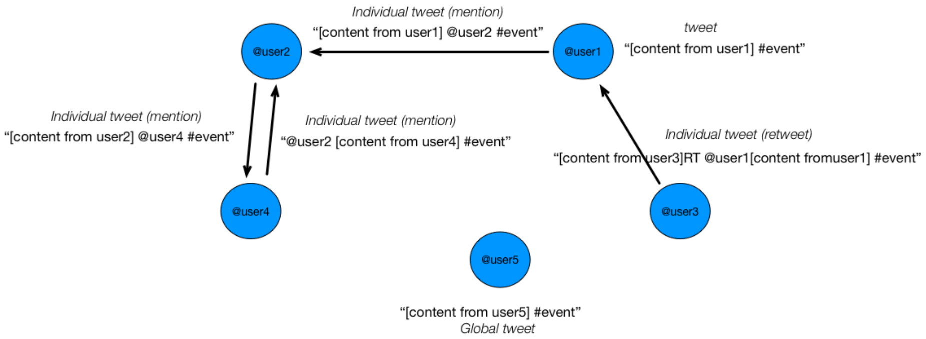

Individual tweet (mention)
“[content from user1] @user2 #event”
@user2
@user1
tweet
“[content from user1] #event”
Individual tweet (mention)
“[content from user2] @user4 #event”
Individual tweet (mention)
“@user2 [content from user4] #event”
Individual tweet (retweet)
“[content from user3]RT @user1[content fromuser1] #event”
@user4
@user3
@user5
“[content from user5] #event”
Global tweet

Fig. 2. Representation of the network based on users' activity in an event in Twitter on-line social network.

Toolkit[5]. Gephi Toolkit is a software library that facilitates the modeling, analysis, and visualization of data that can be represented as a network.

The *Analyzer* agent offers two types of analysis of users' interactions:

- **Network Level Analysis.** At the *network level*, the *Analyzer* agent studies the evolution of the following properties over time: (i) type of interactions (i.e., the number of the different types of messages that were generated in an event); (ii) users (i.e., the evolution of users' participation during an event(s)); (iv) links (i.e., how information flows in the network and influences the formation of the giant component); (v) symmetric links (i.e., in which moment there are conversations between users); (vi) distribution of the degree of connection to understand the topology of the network and how it changes or when it remains constant as time passes; (vii) average path length and diameter to understand in which moment there is a change in the network structure and, therefore in the communication; (viii) clustering (i.e., how interactions among neighbors evolve).
- **Individual Level Analysis**. The *Analyzer* agent studies the evolution of centrality properties of individual users (i.e., network nodes) in one or several events. The centrality properties determine which users are the most important ones based on their location in the network structure. The *Analyzer* agent considers the following centrality metrics: betweenness (i.e., the number of times a user acted as bridge/broker through the shortest path between two other users); in-degree (i.e., indicates whether or not the user was meaningful for other users); out-degree (i.e., indicates the activity of the user); and, eigenvector (i.e., if the users around a node **A** are influential, it makes **A** more influential also). Once the analysis has been performed, it is stored for a period of time in the database just in case the analysis is subsequently requested again.

– *User* agent acts as a mediator agent between users and the system through a web interface. The interface facilitates the interaction through the user and the system

[5] https://gephi.org/toolkit/

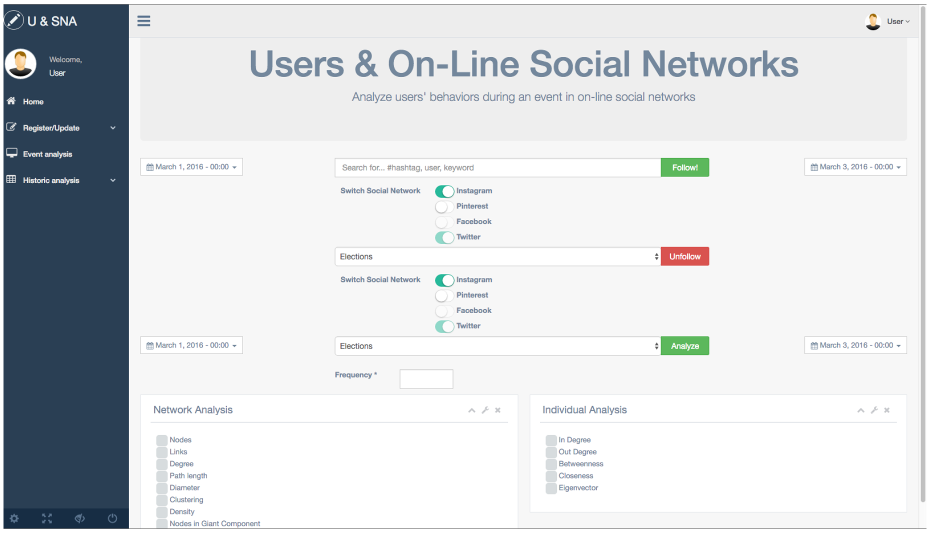


Fig. 3. Web interface that *User* agent provides to facilitate the interaction between users and the MAS framework.

(see Figure 3). The main page offers three possible actions to the users: *follow*, *unfollow*, or *analyze* an event. To *follow* an event, the user introduces an event identifier (i.e., hashtag, keyword, user name), the on-line social networks in which he/she is interested in, and the period of time to follow the event. To *unfollow* an event, the user has a dropdown menu with all the events that he/she has followed. To *analyze* an event, the user should provide the keyword of the event, the time interval, the frequency of the analysis, and the metrics to be considered for the analysis.

3.1 MAS Framework Workflows

Agents in the system interact with each other in order to collect, process, and analyze what is happening in an event in one or several on-line social networks. The system dynamics are shown in the following interaction scenarios: **Follow Event**, **Unfollow Event**, and **Request Analysis**.

The **Follow Event** scenario starts when the agent that represents the user sends a *Request* message to the *Manager* agent in order to start following an event (see Figure 4). The message contains: a keyword that identifies the event, the social networks that the user is interested in, and the time interval the user wants to consider. The *Manager* agent checks whether the event is already being monitored. If the event is being monitored, the *Manager* checks whether there are *Monitor* agents retrieving user activities from the required social networks. Moreover, *Manager* controls if there are enough *Monitor* agents to collect all the data generated during the event. If the event does not exist in the system (i.e., there are no *Monitors* following the event), the *Manager* creates at least one *Monitor* and *Analyzer* agents per each social network and event requested by the user. If the current *Monitor* agents that are in the system are overloaded by the amount of information generated during an event, the *Manager* agent will create more *Monitor* agents in order to deal with the current amount of information. Each *Monitor* agent will store the collected messages in a database.

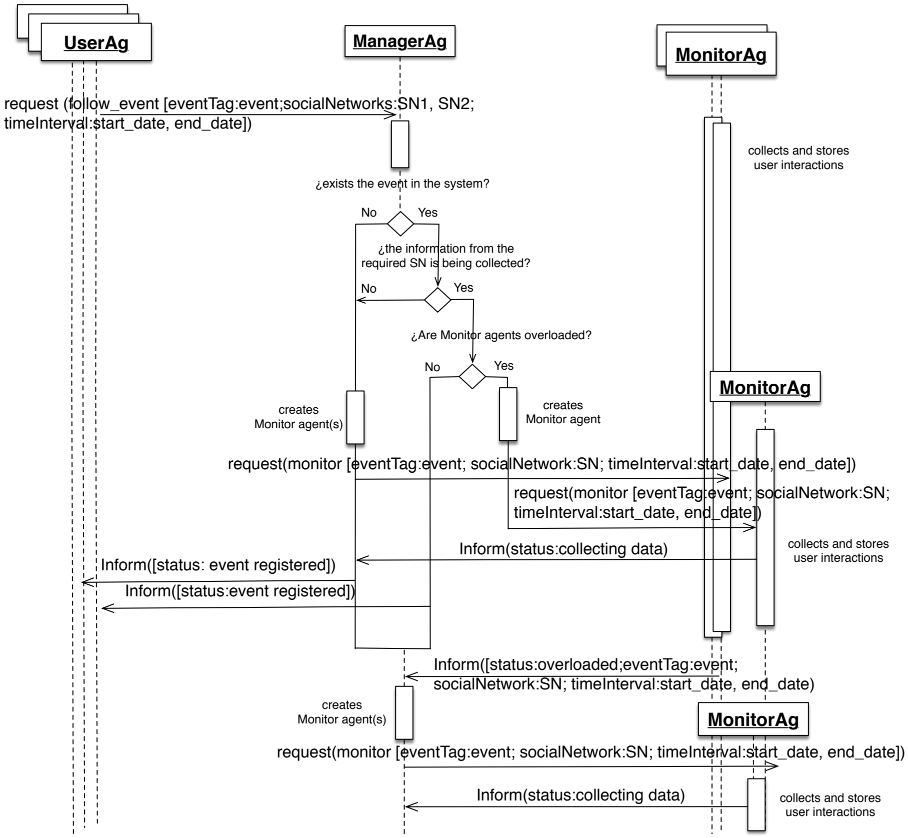

UserAg
ManagerAg
MonitorAg
request (follow_event [eventTag:event;socialNetworks:SN1, SN2;
timeInterval:start_date, end_date])
collects and stores
user interactions
¿exists the event in the system?
No
Yes
¿the information from the
required SN is being collected?
No
Yes
¿Are Monitor agents overloaded?
No
Yes
MonitorAg
creates
Monitor agent(s)
creates
Monitor agent
request(monitor [eventTag:event; socialNetwork:SN; timeInterval:start_date, end_date])
request(monitor [eventTag:event; socialNetwork:SN;
timeInterval:start_date, end_date])
collects and stores
user interactions
Inform(status:collecting data)
Inform([status: event registered])
Inform([status:event registered])
Inform([status:overloaded;eventTag:event;
socialNetwork:SN; timeInterval:start_date, end_date)
creates
Monitor agent(s)
MonitorAg
request(monitor [eventTag:event; socialNetwork:SN; timeInterval:start_date, end_date])
Inform(status:collecting data)
collects and stores
user interactions

Fig. 4. Auml sequence diagram for the *FollowEvent* scenario

A user that was interested in following an event can also request an analysis of the users' activity in the event whenever she/he wants. The **Request Analysis** scenario starts when a user requests an analysis of the event he/she is interested in during a time interval (see Figure 5). First, the *User* agent checks if the time interval for the analysis refers to an event that already happened or is currently active. If it is a past event, the *User* agent checks if the analysis requested by the user is already done. In that case, the *User* agent collects and shows the results of the analysis to the user through the web interface. Otherwise, the *User* agent sends a request message to an *Analyzer* agent.

The message specifies: the event to be analyzed, the social networks that the user is interested in, the time interval for the analysis, the frequency of the analysis (if is an event that is happening in the moment of the analysis, the frequency is 0), and the type of analysis (i.e., network or individual level). Once the *Analyzer* agent receives a request for the analysis of an event, it considers the size of the network and the time interval to delegate part of the analysis to other *Analyzer* agents or perform the complete analysis itself. In both situations, at the end of the analysis, the corresponding *Analyzer* agents will inform the *User* agent about the results of their analysis. The *User* agent waits for receiving all the partial analysis performed by the *Analyzer* agent(s). When the *User* agent has the complete analysis, it will display the results to the final user through a web interface.

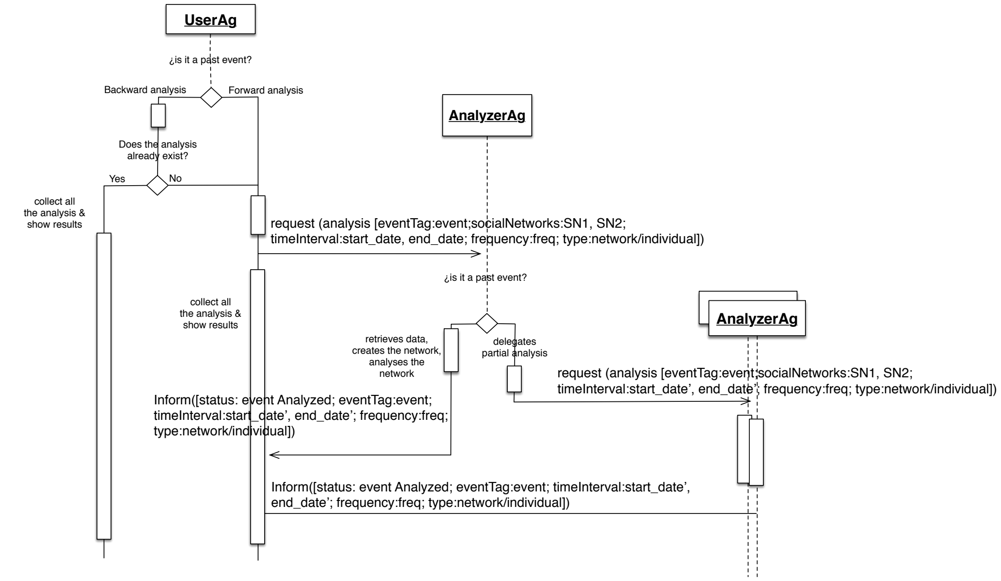

UserAg
¿is it a past event?
Backward analysis
Forward analysis
AnalyzerAg
Does the analysis already exist?
Yes
No
collect all the analysis & show results
request (analysis [eventTag:event;socialNetworks:SN1, SN2; timeInterval:start_date, end_date; frequency:freq; type:network/individual])
¿is it a past event?
collect all the analysis & show results
AnalyzerAg
retrieves data, creates the network, analyses the network
delegates partial analysis
request (analysis [eventTag:event;socialNetworks:SN1, SN2; timeInterval:start_date', end_date'; frequency:freq; type:network/individual])
Inform([status: event Analyzed; eventTag:event; timeInterval:start_date', end_date'; frequency:freq; type:network/individual])
Inform([status: event Analyzed; eventTag:event; timeInterval:start_date', end_date'; frequency:freq; type:network/individual])

Fig. 5. Auml sequence diagram for the *RequestAnalysis* scenario

At some point, a user that was following an event may decide to stop following it (see Figure 6). At that moment is when the **Unfollow Event** scenario starts. Then, the *User* agent sends a request to *Manager* to stop monitoring the event(s). First of all, the *Manager* agent checks if other users are following the event in the system. In case that the event is being followed, the *Manager* checks if it is being followed in the selected social network and if there are more users interested in it or not. If the user is the last one interested in following the event in the selected social network, the *Manager* will send request messages to the *Monitor* agents in order to stop their activity. Then, the manager will inform the *User* agent that the event is not being followed.

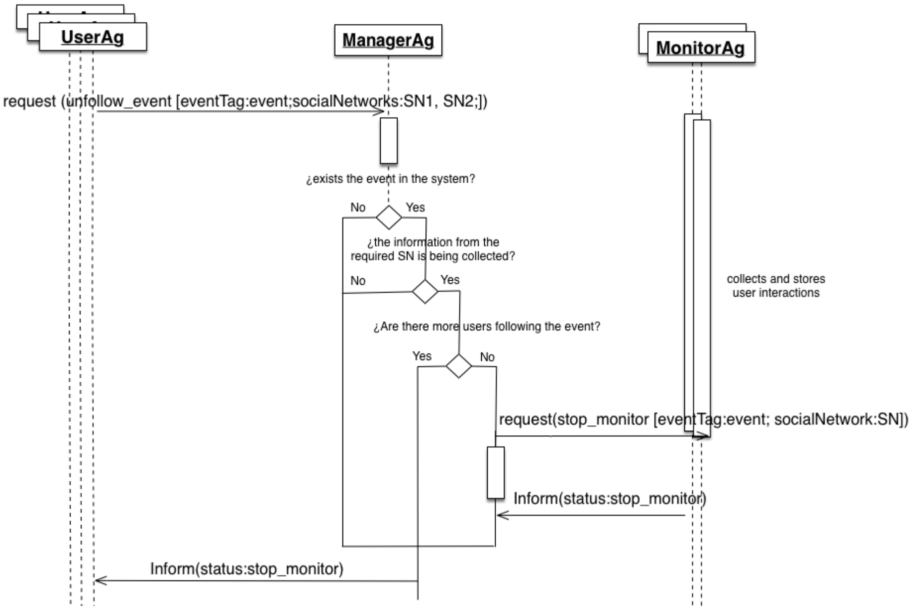
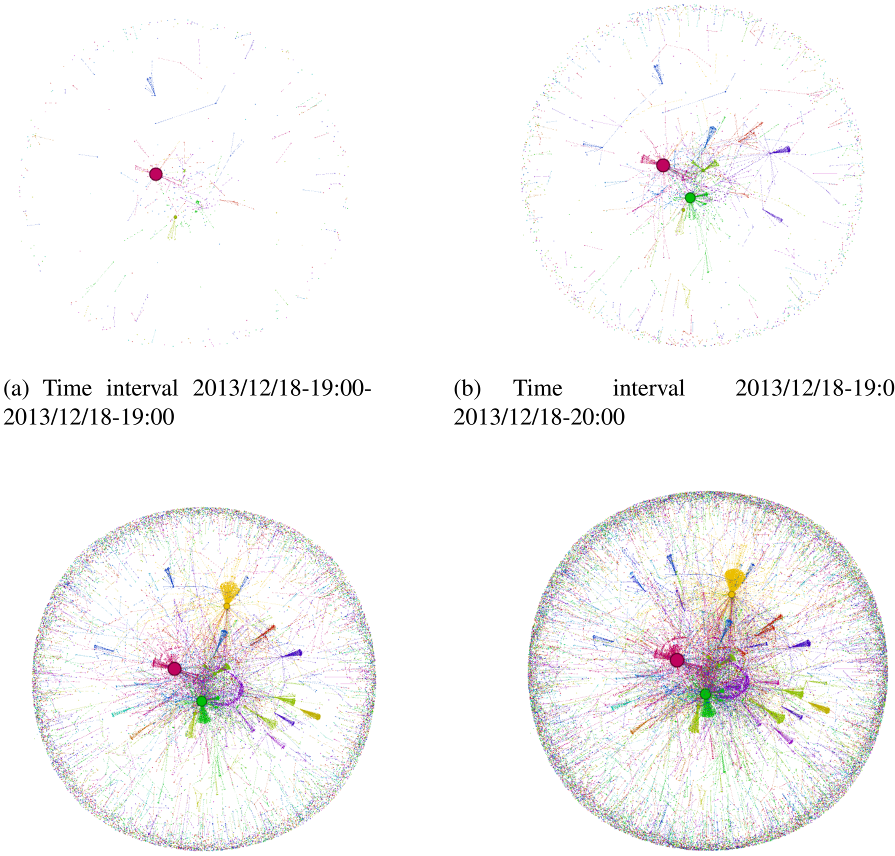

UserAg
ManagerAg
MonitorAg
request (unfollow_event [eventTag:event;socialNetworks:SN1, SN2;])
¿exists the event in the system?
No
Yes
¿the information from the required SN is being collected?
No
Yes
collects and stores user interactions
¿Are there more users following the event?
Yes
No
request(stop_monitor [eventTag:event; socialNetwork:SN])
Inform(status:stop_monitor)
Inform(status:stop_monitor)

Fig. 6. Auml sequence diagram for the *UnfollowEvent* scenario

4 Case study: Analyzing Users' Activity in Twitter Events

The aim of this section is to validate the proposed framework using an specific context that allow us to validate the functionality and the results provided by the framework. We considered a scenario where a user decides to start following a set of events in Twitter in order to analyze the evolution of users' activity in different events. The user interacts with the framework through a web interface (see Figure 3) choosing the keywords and in which on-line social network(s) he/she is interested. Currently, the framework only facilitates the information retrieval from Twitter. Besides that, the user should choose the time interval for collecting users' activity data. Once all this information has been specified, the framework starts to follow users' activity in a transparent way to the user.

The user can request an analysis whenever. The user selects the event to analyze, the period of time, and the frequency of the analysis. Moreover, the user also selects the level of the analysis: *network level* (i.e., metrics that provide information about the global activity), *individual level* (i.e., metrics that provide information about relevant users considering different criteria), or both. The user can select the properties he/she is interested in each level.



(a) Time interval 2013/12/18-19:00-2013/12/18-19:00

(b) Time interval 2013/12/18-19:00-2013/12/18-20:00

(c) Time interval 2013/12/18-19:00-2013/12/18-21:00

(d) Time interval 2013/12/18-19:00-2013/12/18-22:00

Fig. 7. Users interaction network evolution of the event *#LaVoz*. Nodes represent users that publish a message related to the event. Egdes represent interactions between users (i.e., messages that mention other user(s). The size of the node represent its degree of betweenness in the network.)

The framework offers the visualization of users interaction network in different time intervals depending on the period of time specified for the analysis (see Figure 7). The visualization allows users to identify the users that are being more relevant at certain point and see how interactions evolve and how communities emerge. Besides the users interaction network visualization, the framework shows the analysis at *network* and *individual* level.

As an example, we describe the application of the framework presented in Section 3 by means of the analysis of several tv show events. The main features of the final

Event	N	E	clust.	d	path	comp.	k	%GC	%sl
#lavoz	45,914	39,891	0.07	9	2.11	19,398	0.87	39	6.95
#topchef12	26,044	27,155	0.05	25	8.74	10,689	1.04	49.12	12.29
#Operacion Palace	107,606	195,470	0.08	23	8.26	19,504	1.82	80	15.06
#breakingbad	151,473	120,661	0.05	13	3.24	67,060	0.80	47	14.26
#GH15	25,011	56,612	0.08	18	6.71	4950	2.26	75.26	6.17
#GameOfThrones	98,882	96,290	0.09	21	5.45	40,806	0.97	52.6	21.23

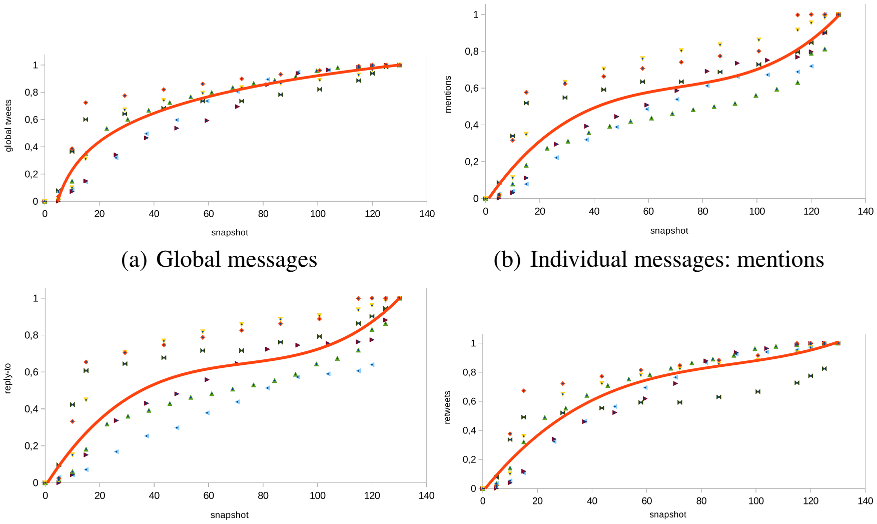
Table 2. High-level statistics in the last snapshot of the social networks analyzed (N = nodes; E = links; clust. = clustering degree; d = network diameter; path = average path length; comp. = number of connected components in the network; k = average degree of connection; %GC = % of nodes in the giant component; %sl = percentage of symmetric links.

network structures of the analyzed events and a brief description of each one are shown in Table 2 and Table 3. The analysis performed was at network and individual level.

Event	Description	Start	End
#lavoz	the last show of a Spanish TV show about music	2013/12/18-20:00:00	2013/12/19-03:00:00
#topchef12	the last show of a Spanish TV show about cooking	2013/12/18-20:30:00	2013/12/19-02:30:00
#Operacion Palace	Spanish news show	2014/02/23-19:30:00	2014/02/24-00:30:00
#breakingbad	last episode of a serial drama	2013/09/30-01:00:00	2013/09/30-06:30:00
#GH15	reality show	2014/12/18-19:30:00	2014/12/10-03:30:00
#GameOfThrones	first episode of a serial drama season	2014/04/07-01:00:00	2014/04/07-06:30:00

Table 3. Description of events on Twitter that have been analyzed.

At *network level*, we selected the analysis of the evolution of the following properties: number of messages, nodes and links, path length and diameter, and percentage of nodes in the giant component (see Figures 8-11). The Y-axis of the graphs shows the value of the property that the agent analyzed. The X-axis shows the time in minutes before, during, and after the event. It is also possible to compare the evolution of different events. In that case, the *Analyzer* agent normalizes the data in order to facilitate the visualization of the evolution of the structural properties of several events that the user is interested in. Each event is represented by a set of points of certain color. In each of



(a) Global messages (b) Individual messages: mentions

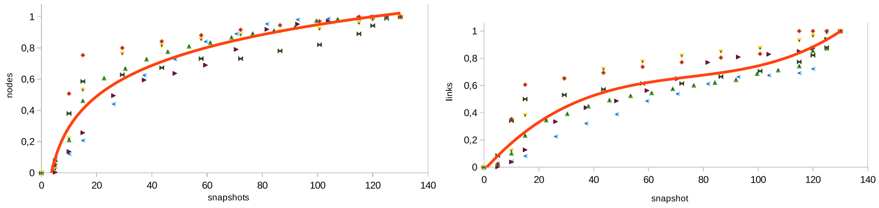
(c) Individual messages: replyTo (d) Individual messages: retweets

Fig. 8. Analysis of the evolution of individual messages of the interaction networks of a set of TV show events.

the properties analyzed, the regression function that best fits the data collected from the events is also shown.

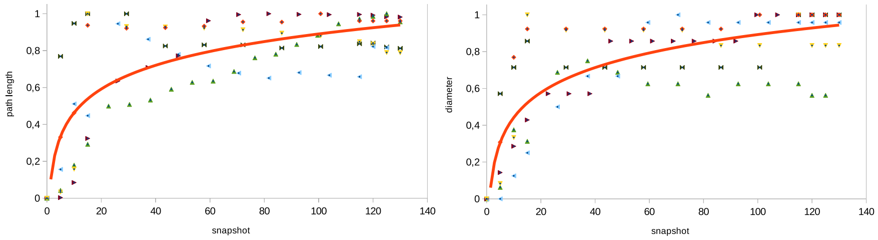
The *number of messages* of each type is not a structural metric but provides insights about what type of interactions occur in an event (see Figure 8). For the number of messages generated by the users in the TV show events analyzed, we observed that there was a difference between the individual messages and global messages. Before the tv shows analyzed, users publish more messages than during and after the event finishes where there is an steady increase in the evolution of global messages. However, in individual messages, there is an important increase before and after the event. This means that users tended to interact with other users after the event in order to share their opinion about what happened during the TV show.

Nodes give us a view of the evolution of users' participation in an event. *Links* allow us to understand if the participation of users in an event is through individual messages, which implies an interaction with other users, what type of interactions occur (mention, retweet, or reply to), and how information flows. In the TV show networks analyzed, the number of nodes and links evolved similarly before and during the event (see Figure 9). Before the TV show started, there was a sharp increase in the number of nodes. This increase continued at a lower rate during the TV show event. At the end of the event, the number of nodes increased steadily and finally remained almost constant. This means that users that participate in the event start participating since the beginning of the event and during and after there were only a few new users. In the case of links, before the TV event started, users started to interact. However, during the event, the number of links continued growing, albeit at a lower rate than before. This fact reflects that the level



(a) Nodes (b) Links

Fig. 9. Analysis of the evolution of nodes and links of the interaction networks of a set of TV show events.



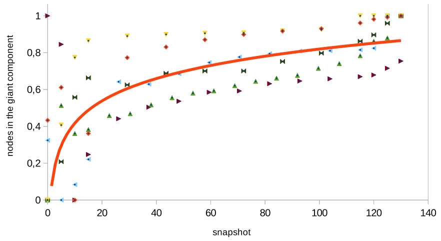
(a) Path (b) Diameter

Fig. 10. Analysis of the evolution of path and diameter of the interaction networks of a set of TV show events.

of interaction decreases during the event. At the end of the event, the last links were individual messages between nodes that were already present in the network since the number of the nodes did not increase. This indicates that at the end of the event nodes interacted with other nodes to comment on what happened during the TV show.

The *average path length* and the *diameter* of the networks increased before the event starts and once the event started the diameter remained almost constant (see Figure 10). The reason for the initial increase was that the majority of the users that participated through individual message usually mentioned an official Twitter account or celebrity and generate a link in one direction (user→celebrity/official account). However, Twitter accounts and celebrities do not respond to the high number of messages received, therefore, there is a high number of unidirectional links. There is a small proportion of users that interacted with each other and there were a few conversations (i.e., bidirectional links). The high number of unidirectional messages with celebrities and official accounts and the small proportion of conversations made the average path and the diameter increase.

In order to determine when is most appropriate to spread information so that the majority of network users can be reached, it is important to know the percentage of the users that belong to the *giant component* (see Figure 11). In general, we have observed that in all the events, the giant component gains the largest number of nodes just before the event starts. Then, there is a moderate increase in the number of nodes that are part



(a) Giant Component

Fig. 11. Analysis of the evolution of the % of nodes that are part of the giant component of the interaction networks of a set of TV show events.

of the giant component. During the events, there is a small proportion of interactions with nodes that are outside the giant component.

The analysis at *individual level* allows the detection of influential users considering different criteria. In order to anonymize data, we classified users in five categories based on their profile: celebrities (i.e., actor/actress, presenter, competitor, jury), official accounts (i.e., TV show/program account or channel account), media (i.e., blog, blogger, radio, journalist), user groups (i.e., fan club), and anonymous users.

The framework allows establishing a ranking of most relevant users considering different criteria. Table 4 shows the number of users' profiles that are among the ten most relevant users following different criteria (i.e., in-degree, out-degree, betweenness, and eigenvector). In general, in TV show events, users that receive higher support (i.e., a higher in-degree) are celebrities and official accounts. The profiles that generated more messages were individual user accounts. This behavior shows that, in type of events, official accounts are missed generating information if you compare with individual user accounts. The users that acted as brokers connecting different communities were user accounts and media. Finally, following the criteria of the eigenvector, the most relevant users were celebrities and and official accounts. This behavior shows that in TV shows official accounts and celebrities usually interact with other similar accounts and not with individual users.

	CELEBRITY		OFFICIAL ACCOUNT		MEDIA		USER		USERGROUP	
	$\overline{x}$	σ	$\overline{x}$	σ	$\overline{x}$	σ	$\overline{x}$	σ	$\overline{x}$	σ
In-degree	4.17	1.47	2.67	1.37	0.83	0,75	2,33	1,03	0,00	0,00
Out-degree	0.33	0.82	0.00	0.84	0,17	0,41	8,17	2,79	0,83	2,04
Betweenness	1.83	0.75	1.33	0.82	2,17	2,40	3,67	1,97	1,00	1,67
Eigenvector	4.33	2.07	2.83	0.00	0,67	0,52	2,17	1,83	0,00	0,00

Table 4. Individual level analysis of a set of TV show events.

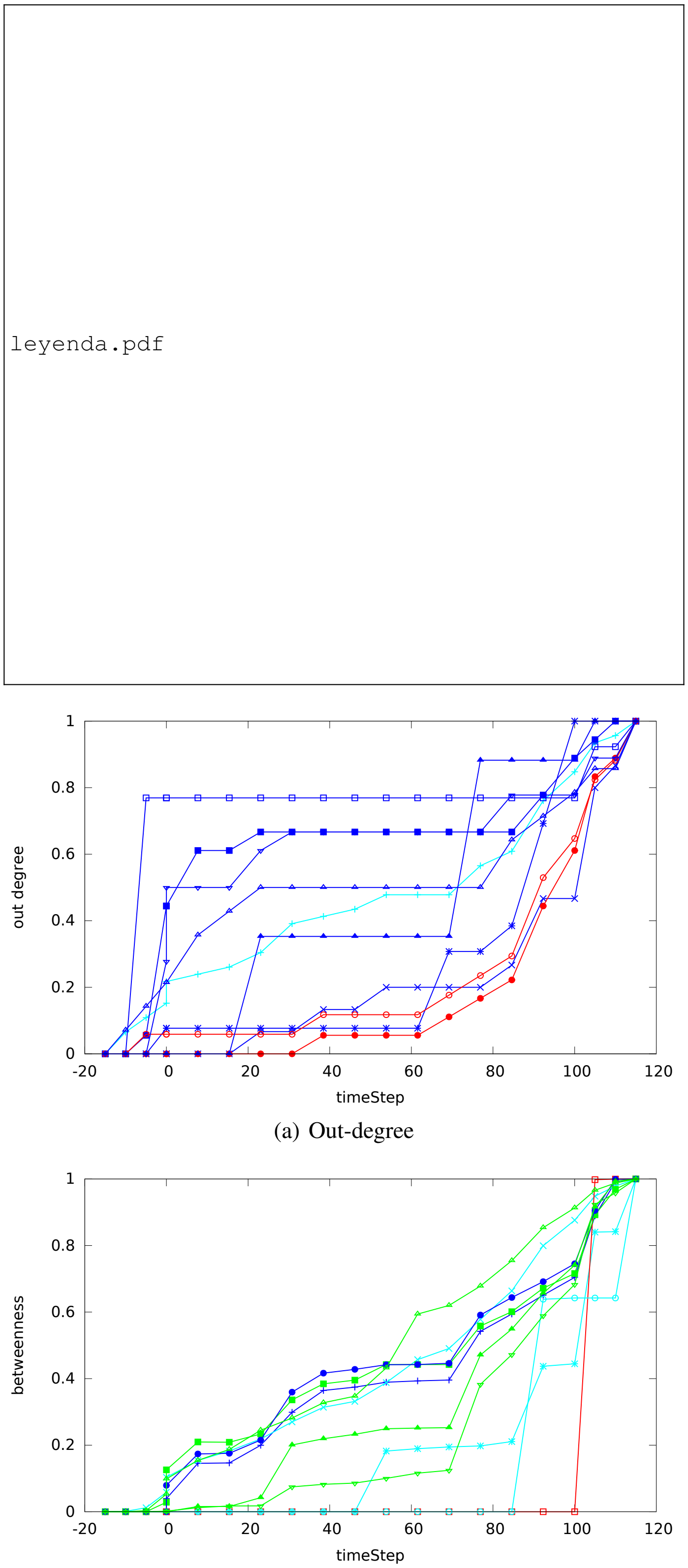
The framework also allows monitoring a certain individuals following some of the criteria mentioned before. This analysis allows users to observe the evolution of the relevance of specific individuals during the event. Figure 12a shows that there are differences between the behavior of users' profiles. Unknown users participate before the event starts. However, celebrities wait until the end of the event to participate generating messages. In Figure 12b, betweenness of media, and users increased before the event but the most important increase occurs at the end of the event. Therefore, the relevance of these profiles plays an important role at the end of the event. According to the betweenness values, if we want to increase the diffusion of a message, it would be better to spread it at the end of the event through individual users with high betweenness.

5 Conclusions

The number of people and organizations using on-line social networks as a new way of communication is continually increasing. There is a great quantity of data that can be used for research on human behavior patterns. The analysis of this information facilitates the identification of communication patterns in on-line media events. It will allow the identification of individuals and groups that play central roles in the diffusion of information. The analysis is also useful to find opportunities to accelerate information flows.

In this paper, we have presented a Multi-Agent Framework for the analysis of the evolution of users' behavior during events in on-line social networks. The system is responsible for collecting, processing, analyzing, and presenting information retrieved from the activities of users in on-line social networks. The use of a multi-agent system for the framework provides intrinsic characteristics of MAS such as flexibility, adaptability, and scalability needed in an environment where there is a variable users' demand and a variable users' activity depending on the event. In the proposed framework, we consider a period of time in the analysis of users' behavior during an event instead of a single point. Moreover, the framework offers a complete vision of what is happening during an event taking into account structural properties of temporally annotated network generated from users' interactions. The system offers an analysis at network level (i.e., offers a view of the global behavior of users during an event) and at individual level (i.e., which users are playing a key role during an event considering different centrality criteria).

The framework was validated in a set of tv show events in Twitter in order to analyze users' activity. At *network level*, the analysis provides insights about how and when users communicate during an event and what type of communication used in each moment. In the analyzed TV show events, the analysis conclude that users tend to participate through global messages. The majority of interactions are unidirectional from unknown users to official accounts or celebrities. This fact is clearly reflected in the structural properties of the networks. The number of nodes is higher than the number of links, which means that users prefer to participate through global messages that do not imply interaction (i.e., the creation of a link), rather than interact with other users. The small proportion of individual messages are mentions. These messages are from anonymous users to a celebrity or an official Twitter account that usually does not respond or interact with anonymous users. This fact is reflected in the low percentage of symmetric links. One of the effects of the lack of conversations (i.e., bidirectional



(a) Out-degree

(b) Betweenness

Fig. 12. Individual level analysis of the evolution of relevance of user profiles #LaVoz considering the betweenness and out-degree criteria.

links) is that the path length and the diameter are not reduced as the number of interactions increases. Another structural property that reflects that there is a low level of social interaction is that users do not interact with other nodes in their neighborhood (there is a low degree of clustering). At *individual level*, the framework allows to identify which type of profiles and/or users are receiving support form other users, who is generating a higher number of messages, who is connecting communities and therefore plays a critical role in the information diffusion, and who is connected with other relevant users. In the case of TV shows, the official Twitter accounts and celebrities have the highest in-degree and betweenness. Therefore, they play a relevant role in the information diffusion process. The nodes with the highest outdegree are anonymous users. Regarding the eigenvector values, in TV show events celebrities are the profiles with highest values. This fact means that they usually interact with other celebrities rather than anonymous users. The connectivity of the interaction network would be improved if official accounts and celebrities would interact more with anonymous users and if these interactions would occur at the beginning of the events.

This analysis provides insights about characterizing and understanding complex interaction structures, how these structures emerge, and how the structure of interactions among users can be improved in events in on-line social networks. As a future work, we plan to include other on-line media in order to provide a more complete analysis on multiplex networks of what is happening during an event.

6 Compliance with Ethical Standards

Funding: This work is partially supported by the PROMETEOII/2013/019, TIN2014-55206-R, TIN2015-65515-C4-1-R, H2020-ICT-2015-688095.

Conflict of Interest: Elena del Val declares that she has no conflict of interest. Carla Martinez declares that she has no conflict of interest. Vicente Botti declares that he has no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors. All the data used in the experiments were obtained from public datasets.

References

1. R. Abdulrahman, D. Neagu, and D. Holton. Multi agent system for historical information retrieval from online social networks. In *Agent and Multi-Agent Systems: Technologies and Applications*, volume 6682, pages 54–63. 2011.
2. Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong. Analysis of topological characteristics of huge online social networking services. In *Proceedings of the 16th WWW*, pages 835–844, 2007.
3. S. Bastiaensens, H. Vandebosch, K. Poels, K. V. Cleemput, A. DeSmet, and I. D. Bourdeaudhuij. Cyberbullying on social network sites. an experimental study into behavioural intentions to help the victim or reinforce the bully. *Computers in Human Behavior*, 31(0):259 – 271, 2014.

4. F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing user behavior in online social networks. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 49–62. ACM, 2009.
5. F. Bergenti, E. Franchi, and A. Poggi. Selected models for agent-based simulation of social networks. In *3rd Symposium on Social Networks and Multiagent Systems (SNAMAS 2011)*, pages 27–32, 2011.
6. J. Borge-Holthoefer, A. Rivero, I. García, E. Cauhé, A. Ferrer, D. Ferrer, D. Francos, D. Iñiguez, M. P. Pérez, G. Ruiz, et al. Structural and dynamical patterns on online social networks: the spanish may 15th movement as a case study. *PLoS One*, 6(8), 2011.
7. J. Borondo, A. J. Morales, J. C. Losada, and R. M. Benito. Characterizing and modeling an electoral campaign in the context of twitter: 2011 spanish presidential election as a case study. 2013.
8. S. A. Catanese, P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti. Crawling facebook for social network analysis purposes. In *Proceedings of the international conference on web intelligence, mining and semantics*, page 52. ACM, 2011.
9. M. Cha, A. Mislove, and K. P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th international conference on World wide web*, pages 721–730. ACM, 2009.
10. D. H. Chau, S. Pandit, S. Wang, and C. Faloutsos. Parallel crawling for online social networks. In *Proceedings of the 16th WWW*, pages 1283–1284. ACM, 2007.
11. E. del Val, C. Martínez, and V. Botti. A multi-agent framework for the analysis of users behavior over time in on-line social networks. In *10th International Conference on Soft Computing Models in Industrial and Environmental Applications*, pages 191–201. Springer, 2015.
12. E. del Val, M. Rebollo, and V. Botti. Does the type of event influence how user interactions evolve on twitter? *PLOS one*, 10(5):e0124049, 2015.
13. S. Esparcia, E. Argente, V. Julián, and V. Botti. Gormas: A methodological guideline for organizational-oriented open mas. In *Handbook on Agent-Oriented Design Processes*, pages 173–218. 2014.
14. Eurostat. Internet use statistics - individuals. http://ec.europa.eu/eurostat/statistics-explained/index.php/Internet_use_statistics_-_individuals, Feb 2016. [Online; accessed 29-April-2016].
15. Eurostat. Social media - Statistics on the use by enterprises. http://ec.europa.eu/eurostat/statistics-explained/index.php/Social_media_-_statistics_on_the_use_by_enterprises#Further_Eurostat_information, Feb 2016. [Online; accessed 29-April-2016].
16. A. M. García Fornes, M. RODRIGO SOLAZ, A. M. Terrasa Barrena, J. Inglada, V. Javier, J. Jorge Cano, L. MULET MENGUAL, A. Palomares Chust, L. A. BÚRDALO RAPA, A. S. Giret Boggino, et al. Magentix 2 user's manual. 2015.
17. M. Gatti, P. Cavalin, S. B. Neto, C. Pinhanez, C. dos Santos, D. Gribel, and A. P. Appel. Large-scale multi-agent-based modeling and simulation of microblogging-based online social network. In *Multi-Agent-Based Simulation XIV*, pages 17–33. 2014.
18. J. Golbeck, C. Robles, and K. Turner. Predicting personality with social media. In *CHI '11*, pages 253–262, 2011.
19. R. Guimerà, A. Llorente, E. Moro, and M. Sales-Pardo. Predicting human preferences using the block structure of complex social networks. *PloS one*, 7(9):e44620, 2012.
20. B. A. Huberman, D. M. Romero, and F. Wu. Social networks that matter: Twitter under the microscope. *arXiv preprint arXiv:0812.1045*, 2008.
21. M. Jamali and H. Abolhassani. Different aspects of social network analysis. In *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, pages 66–72. IEEE, 2006.

22. Y. Jiang and J. Jiang. Understanding social networks from a multiagent perspective. *Parallel and Distributed Systems, IEEE Transactions on*, 25(10):2743–2759, 2014.
23. M. Klusch. *Intelligent information agents: agent-based information discovery and management on the Internet*. Springer Science & Business Media, 2012.
24. G. Kossinets and D. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006.
25. R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *Link mining: models, algorithms, and applications*, pages 337–357. Springer, 2010.
26. D. Lazer. Life in the network: the coming age of computational social science. *Science (New York, NY)*, 323(5915):721–723, 2009.
27. J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), May 2007.
28. C. Licoppe and Z. Smoreda. Are social networks technologically embedded?: How networks are changing today with changes in communication technology. *Social networks*, 27(4):317–335, 2005.
29. G. Lotan, E. Graeff, M. Ananny, D. Gaffney, I. Pearce, and D. Boyd. The revolutions were tweeted: Information flows during the 2011 tunisian and egyptian revolutions. *International Journal of Communication*, 5:1375–1405, 2011.
30. A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proc. of the 7th ACM SIGCOMM*, pages 29–42, 2007.
31. A. Morales, J. Losada, and R. Benito. Structure and dynamics of emerging social networks from twitter's conversation. *Int. J. Complex Systems in Science*, 1(2):216–220, 2011.
32. A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf. *Coordination of Internet agents: Models, technologies, and applications*. Springer Science & Business Media, 2013.
33. I. Peña-López, M. Congosto, and P. Aragón. Spanish indignados and the evolution of 15m: towards networked para-institutions. *Big Data: Challenges and Opportunities*, pages 25–26, 2013.
34. A. Perliger and A. Pedahzur. Social network analysis in the study of terrorism and political violence. *PS: Political Science & Politics*, 44:45–50, 1 2011.
35. D. M. Romero, W. Galuba, S. Asur, and B. A. Huberman. Influence and passivity in social media. In *Proceedings of the 20th WWW*, pages 113–114, 2011.
36. D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In *Proceedings of the 20th WWW*, pages 695–704, 2011.
37. F. N. Stockman and P. Doreian. Evolution of social networks: Processes and principles. In *Evolution of Social Networks*, pages 233–250. Routledge, 1997.
38. C. Tekin, S. Zhang, and M. van der Schaar. Distributed online learning in social recommender systems. *Selected Topics in Signal Processing, IEEE Journal of*, 8(4):638–652, 2014.
39. A. L. Traud, P. J. Mucha, and M. A. Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.
40. J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, 2011.
41. S. Valero, E. del Val, J. Alemany, and V. Botti. Using magentix2 in smart-home environments. In *10th International Conference on Soft Computing Models in Industrial and Environmental Applications*, pages 27–37. Springer, 2015.
42. S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
43. Wersm. How Much Data Is Generated Every Minute On Social Media? http://wersm.com/how-much-data-is-generated-every-minute-on-social-media/, Aug 2015. [Online; accessed 29-April-2016].