MASIR: A Multi-agent System for Real-Time Information Retrieval from Microblogs During Unexpected Events

Imen Bizid, Patrice Boursier, Jacques Morcos, Sami Faiz

MASIR : A Multi-agent System for Real-time Information Retrieval from Microblogs during Unexpected Events

Imen Bizid[1,2], Patrice Boursier[2,3], Jacques Morcos[2], and Sami Faiz[1]

[1] LTSIRS Laboratory, Tunis, Tunisia
sami.faiz@insat.rnu.tn
[2] L3i Laboratory, University of La Rochelle, La Rochelle, France
{imen.bizid,patrice.boursier,jacques.morcos}@univ-lr.fr
[3] IUMW, Kuala Lumpur, Malaysia
patrice@iumw.edu.my

Abstract. Microblogs have proved their potential to attract people from all over the world to express voluntarily what is happening around them during unexpected events. However, retrieving relevant information from the huge amount of data shared in real time in these microblogs remain complex. This paper proposes a new system named MASIR for real-time information retrieval from microblogs during unexpected events. MASIR is based on a decentralized and collaborative multi-agent approach analyzing the profiles of users interested in a given event in order to detect the most prominent ones that have to be tracked in real time. Real time monitoring of these users enables a direct access to valuable fresh information. Our experiments shows that MASIR simplifies the real-time detection and tracking of the most prominent users by exploring both the old and fresh information shared during the event and outperforms the standard centrality measures by using a time-sensitive ranking model.

Keywords: multi-agent systems, microblogs, real-time, information retrieval, unexpected events

1 Introduction

Since the emergence of Twitter, microblogs have experienced a new social aspect allowing users to express in real time their opinions, perceptions and expectations. The usefulness of such networks has been demonstrated during the Arab Spring revolutions and the last memorable natural disasters. Microblogs were the main source of information used by official organizations and news channels in the different event phases (i.e. event detection, situation awareness, alert dissemination) [3]. Since, these online social platforms have become indispensable to acquire and monitor information in real time during unforeseen events.

Unexpected events, in the context of this paper, refer to the events which occur without any warning or preparation such as natural and human disasters. Information retrieval during this kind of events is more complex than during

simple ones as it is not possible to gain a priori information before its occurrence. Most of the proposed IR approaches in this context are based on both machine and human intelligence [6]. Such techniques are time consuming and require the presence of an important number of volunteers for the collection and the annotation of relevant information tasks.

Indeed, to enhance situation awareness of such events, it is important to find new ways coping with the limited data access using microblogs APIs on the one hand and to be able to retrieve valuable information shared in real time on the other hand [8]. Identifying and tracking prominent users who are behind the relevant and exclusive information shared during the targeted event can offer a rapid access to the required information. The prominence of users in these events relies on the relevance and freshness of their information independently of their social influence. Hence, using traditional techniques would fail to track and identify such users in real time due to the strict microblogs conditions for tracking users and the complexity of this type of events. The original contributions of this paper are twofold. First, we propose a multi-agent system to insure a real-time identification and an extensive tracking of prominent users in the context of managing unexpected events. Second, we propose a novel approach to detect prominent users based on both their geographic and social positions over time during an unexpected event.

This paper is organized as follows: Section 2 reviews existing approaches for information retrieval from social networks. Section 3 describes our proposed system and its functional model. Section 4 presents our experiments and evaluates our results. Section 5 summarizes our conclusions and future work.

2 Related Work

Tracking of social media users for information retrieval has been studied using mainly three ways: social networks' interface crawling through public and false profiles and phishing techniques [4], social networks applications such as Netvizz [10], as well as social networks Application Programming Interfaces (APIs) [1]. These techniques have provided a direct access to an important number of users. However, they have no selection strategy to evaluate the prominence of tracked users.

To the best of our knowledge, the issue of prominent users identification has never been explored in the context of unexpected events. However, there have been several attempts proposing new measures identifying influential users and domain experts in microblogs and specially Twitter [7, 12, 13]. Most of the proposed approaches identifying social media influencers are based on standard centrality measures such as eigenvector centrality and its variants HITS [2] and PageRank[7]. These adapted measures to microblogs specificities (e.g. number of tweets, mentions, retweets ...) have yielded promising results for the identification of influential users [11, 14]. However, they are computationally expensive and sensitive to well-connected users (e.g. celebrities, communication channels...) [5]. On the other hand, low research works have proposed domain experts iden-

tification based on efficient ranking models using a set of features describing the activities of users in particular topics [9, 15]. These approaches are efficient to identify in real time topical authorities sharing relevant information based on their previously shared information.
While somewhat similar to [9, 15], prominent users identification in the context of unexpected events differs in different points. Firstly, there is no a priori information describing the user's prominence that may be extracted before the occurrence of the event. In our context, information related to interested users in an event has to be explored in real time. Secondly, many features describing the prominence of a user in specific events have never been explored in the prior approaches such as the geolocation position, the time of the first user's activity regarding the event and the social position of the user in both the network and the event. Thirdly, our approach is tested in a real case from the identification of all users interested in the event to the detection and the tracking of the most prominent ones.
The problem of prominent users identification has been studied only theoretically using test databases. Microblogs APIs constraints related to extensive crawling have never been taken into account. Moreover, existing approaches for microblogs users tracking have focused only on the quantity of users while neglecting their qualities. In this paper, we propose a multi-agent system for relevant and exclusive information retrieval in Twitter by identifying and tracking the most prominent users in real time cases.

3 Decentralized Collaborative System for Information Retrieval

This section describes the operation details of MASIR for information retrieval from microblogs during unexpected events. MASIR is based on a multi-agent approach compliant with Twitter APIs specifications. The idea behind modeling a multi-agent system for information retrieval relies on its ability to retrieve information using parallel processing which makes the analysis process computationally feasible in real time. Moreover, the multiple used agents boost the number of monitored users in parallel and ease the detection of the most prominent users during the event.

Fig. 1 describes the decentralized structure of our system. MASIR is composed of 6 different kinds of agents designed to execute a well-defined related tasks. The process starts when the keywords and/or hashtags representing the targeted event were specified to the Stream Retrieval Agent (SRA). Using these parameters, SRA searches for the list of new users sharing real-time information about the event and sends it continuously to the Historic Listener Agents Manager (HLAM). HLAM assigns a Historic Listener Agent (HLA) to each identified user by SRA in order to extract and store his historic in the Historic and Social Information Base (HSIB). The collected data is analyzed by the Prominent Users Detector (PUD) in order to detect the most prominent users that have to

be tracked in real time by the Stream Listener Agents (SLAs). These agents are described in detail in the following sub-sections.

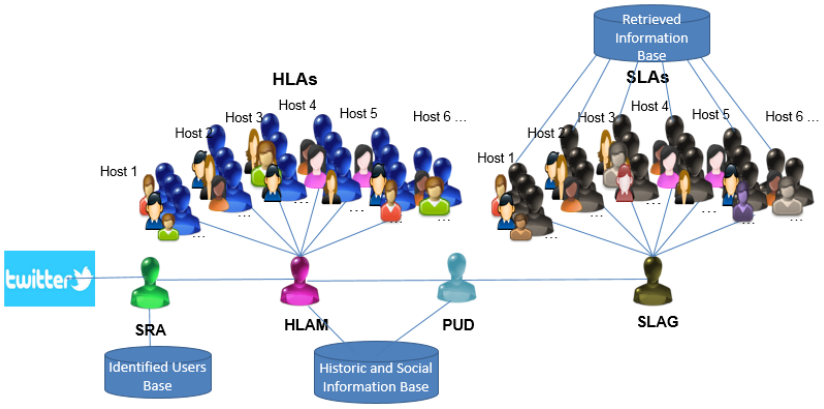

Fig. 1. A decentralized multi-agent system for real-time information retrieval from Twitter

3.1 The Stream Retrieval Agent (SRA)

SRA retrieves the tweets published in real time about an event and extracts the identities of users who are sharing it by following these monitoring operations:

1. *Streaming search:* SRA remains connected to Twitter during the event in order to search in real time for new tweets using the assigned hashtags or keywords identifying the targeted event.
2. *Users' identification:* SRA extracts the identity of users sharing on-topic tweets.
3. *Users' filter:* SRA applies dynamically new filters with reference to the Identified User Base (IUB) in order to force the streaming search to retrieve only tweets shared by new users.
4. *Users' storage:* SRA stores in IUB the identifier of any new detected user posting information related to the event.
5. *List of users sending:* SRA has to send the list of new detected users interested in the event every 30 seconds to HLAM.

3.2 The Historic Listener Agents Manager (HLAM)

HLAM manages the extraction process of the social and historic information from the identified users profiles. It controls multiple HLA agents which are in charge of the historic and social information collection from each identified

user profile. HLAM can undergo different transitions according to the processed operations:

1. *Users assignment:* When HLAM receives the list of users sent by SRA, it adds the new identified users in a waiting list. Then, HLAM assigns each user to one of the available HLAs by respecting the FIFO (First In First Out) principle.
2. *Information reception:* This operation is processed after the reception of a message from HLA precising that the historic extraction process was accomplished. Then, at this stage HLAM saves the returned information collected by HLA in HSIB.
3. *HLA status change:* Once HLAM has received all the extracted information from a HLA, it sets this HLA status to "free" in order to be able to assign it to a new user.

3.3 The Historic Listener Agents (HLAs)

HLAs have to extract historic information shared by each assigned user. Once a HLA has finished the extraction of the needed information related to a user, it sends a message to HLAM to store the collected information in HSIB. Then, the HLAM will change this HLA status to "free". Each HLA have to be able to process the three following operations:

1. *Receiving a user's identity:* When HLA status is set to "free", HLA could be assigned to a unique user recognized by his unique identifier.
2. *Historic information extraction:* HLA extracts all the historic information shared by the assigned user.
3. *Social information extraction:* HLA extracts the list of the followers and followees of the assigned users.
4. *Extracted information Sending:* HLA sends all the information collected to HLAM in order to store it in HSIB and to change its status.

3.4 The Prominent Users Detector (PUD)

PUD acts as the intermediary between the historic extraction process and the streaming process. This agent detects the most prominent users using the data collected during the historic extraction process. The identification of these prominent users insures the attribution of the limited number of parallel SLAs to the most central users during the event. PUD detects the most prominent users by calculating and updating periodically the Prominence Score (PS) of the already watched users. This final score is estimated according to the geo-location and social positions of the user and the recency of his first interaction regarding the event. PS is computed using the following ranking model:

$$PS(u) = w_1 * RS(u) + w_2 * GPS(u) + SPS(u) \quad (1)$$

Where 0.38 and 0.02 are the weights reflecting the importance of RS and GPS. All the weights' values (from w_1 to w_6) used during the calculation of PS and

SPS were estimated a priori through a user study evaluating the active Twitter users in the South Korea ferry disaster. This study was conducted by a group of volunteers who have evaluated the Twitter users according to the relevance and recency of their information about the disaster. These volunteers have noted these users from 1 to 10 according to their prominence. These notes were used for fitting a linear regression model composed of the different predictor scores proposed in this paper to evaluate the prominence of each user. The weights evaluating each predictor were normalized to form the sum 1 for all the weights.
The Recency Score (RS) indicates the recency of the first on-topic information shared by the user regarding the time of occurrence of the event (t_{event}). The difference in time between the first shared information (t) and (t_{event}) is measured in minutes.

$$RS(u) = \frac{1}{t - t_{event} + 1} \tag{2}$$

The Geo-location Position Score (GPS) indicates the inclusion rate of the geo-location(i.e. longitude, latitude) specified by the user in the territory concerned by the event. The event area is represented by a polygon or a set of polygons (Pe) that may include many distant zones. For each user u, we extract from his different historic tweets collected by HLAs the set of his geo-locations (Cu). For example, if all the geolocations specified by the user are included in the event area, his GPS will be set to 1.

$$GP(u) = \frac{Cu \cap Pe}{Cu \cup Pe} \tag{3}$$

The Social Position Score (SPS) indicates how much the user's followers (F) and followees (Fe) are interested in the analyzed event. The more a user has many on-topic followers (OnF) and followees ($OnFe$) having a high RS, the more his final SPS is high. As well-connected users such as CNN and BBC may have a large number of OnF and $OnFe$ even they are not sufficiently prominent regarding the analyzed event, these numbers are adjusted by F and Fe which makes our SPS insensitive to well connected users. SPS is computed as follows using the social information already extracted by HLA and stored in HSIB:

$$SP(u) = w_3 * \frac{\sum_{i=1}^{OnF} RS(i)}{log(OnF+1)} + w_4 * \frac{OnF}{log(F)} + w_5 * \frac{\sum_{i=1}^{OnFe} RS(i)}{log(OnFe+1)} + w_6 * \frac{OnFe}{log(Fe)} \tag{4}$$

Where $w_3 = 0.21$, $w_4 = 0.1$, $w_5 = 0.23$ and $w_6 = 0.04$ are the weights reflecting the importance of the different predictors contributing in the final SPS score of each user.

3.5 The Stream Listeners' Agents Generator (SLAG)

SLAG manages the tracking process of the most prominent users during the event. It starts the generation and management process when it receives the list of prominent users by PUD. SLAG generates a SLA for each user in the list. These SLAs are generated in different hosts in order to avoid the risk of IP banning by Twitter. Hence, SLAG processes the following operations :

1. *Receiving detected users:* SLAG receives periodically an updated list of the most prominent users that have to be tracked in real time.
2. *Killing existing SLAs:* After receiving the updated list, LAG kills SLAs which are tracking users who do not exist in the new list. By killing these SLAs, SLAG will release the place in some hosts in order to be able to track the new prominent users.
3. *Generating a new SLA:* After liberating the place for the new prominent users, SLAG generates SLAs for these users.

3.6 Streaming Listener Agents (SLAs)

While HLA extracts the historic and social information of one assigned user and once it has finished this task, it listens to another user, SLAs differ in various points. First, SLAs keep listening to a user profile in order to detect any new update. Second, SLAs are dynamically generated by the SLAG. Each SLA is in charge of tracking the assigned user profile in real time. SLAs store in real time any new detected information shared by its assigned user in the Retrieved Information Base (RIB). RIB contains the tweets extracted in real time from the most prominent users' profiles.

4 Experiments and Results

The architecture of MASIR was implemented using Java Agent DEvelopment framework (JADE). Using this framework, each agent was created in a running instance named container. MASIR agents were executed in various containers distributed in 5 hosts connected via a Virtual Private Network. Our system was implemented using two Twitter APIs; the Search API for the historic information extraction process and the Streaming API for the real-time tracking of prominent users.

As the Streaming and Search APIs limit the number of the crawled profiles simultaneously to around 5, MASIR used to encounter this limit by distributing SLAs and HLAs in various hosts and by using different Twitter accounts. This distribution aims not only to avoid IP banning when the authorized crawling limit rate is reached but also to boost the number of listened profiles. The 5 used hosts incorporate all a main container in order to enable SLAG to monitor all the created agents and automate the agents generation process according to the number of available hosts. Using this strategy, HLAM managed up to 75 HLAs (15/host) and SLAG generated up to 175 SLAs (35/host).

MASIR was run during the Herault floods which have occurred in the Herault country in France. These floods have lasted two days from 29 to 30 September 2014. They have caused important damages estimated between 500 and 600 million Euros. MASIR was launched after a while from the official announcement of the event using the hashtag "Herault" which was used by Twitter users to refer to the event. Our system has collected 41.064 historic tweets and 22.136 fresh tweets shared respectively by 3.143 users listened by HLAs and 604 users tracked

by SLAs. Hence, MASIR has coped with the limits imposed by the Twitter APIs by tracking an important number of users in real time.
In order to evaluate the quality of the users tracked in real time by MASIR. We have collected all the tweets shared by the identified users by SRA from the announcement of the event to its end using the Search API after two days of the disaster. Then, these users were evaluated by a group of volunteers to define our ground-truth. This group were asked to note these users from 1 to 10 according to the relevance and freshness of their tweets. The top 175 users were selected in order to evaluate the users tracked by MASEA periodically.
Table 1 presents the total number of prominent and non prominent users identified by MASIR during the two days of the disaster and the number of the true prominent users tracked at each period of time with reference to the ground truth results.
According to these results, an important number of the ground-truth prominent users were identified by SRA and tracked by SLAs from the first day of the disaster. We also note that the precision of our detection process was improved during the end of the second day by tracking 46% of the ground-truth prominent users continuously. We compared the used model by MASIR to detect prominent

Table 1. The simulation results of the identified and the true detected users by MASIR

		Identified users by SRA	Ground-truth prominent users	True prominent users listened by SLAs
1st day	12 am-00 pm	1254	157	67
	00 pm-12 am	1264	157	67
2nd day	12 am-00 pm	2433	173	57
	00 pm-12 am	3143	175	81

users with three baseline algorithms: the eigenvector centrality, PageRank and HITS algorithms. These measures were chosen as they were used in the literature for the detection of prominent users in various contexts.
To measure the quality of results returned by each baseline in each period of time, we calculated the precision of the returned prominent users by each algorithm. The obtained results are shown in Fig.2.
Compared with the time consuming centrality measures, our model gains a significant increase in performance at the different stages of the event. We also note that the performance of the baseline measures decreases over time as they are sensitive to well-connected users. Moreover, we note that the centrality measures are not suitable for the detection of prominent users in the context of unexpected events.
According to the results of our experiments, MASIR outperforms the centrality

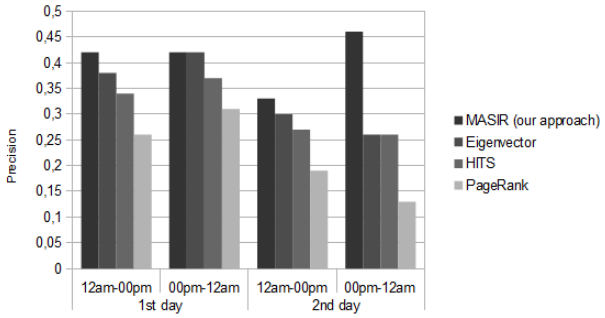

Fig. 2. The precision of *MASIR* vs *Eigenvector, Hits* and *PageRank* during the disaster

measures in terms of performance and time as the most prominent users were detected at an early stage of the event. Moreover, the used distributed architecture and our time sensitive ranking model have made the detection and the tracking of these users feasible in real time. Even MASIR has not detected all the top 175 users, these results are promising. Similar to recommendation and information retrieval engines, we can argue that if from the 9 recommended users, 8 are bad and 1 is extremely good and is sharing the required information, there is a high chance that the user will be satisfied by the retrieved information in real time.

5 Conclusion and Future Work

This paper highlights the power of multi-agent systems based architecture for real-time information retrieval from microblogs during unexpected events. MASIR uses various collaborative agents enabling a real-time detection of the most prominent users who tend to share valuable information. This first research effort to deal with the detection and tracking of prominent users in real unexpected events cases has provided promising results. The straightforward time sensitive measures used by our ranking model have outperformed the standard centrality measures. Moreover, the employed multi-agent architecture has coped with the Twitter APIs limits to be able to track in real time 175 users using only 5 hosts.

For future work, we aim to propose new features reflecting the user behavior during the event in order to improve the detection process of prominent users. In addition, we would like to optimize the distribution of our agents and minimize the number of exchanged messages between the different agents.

References

1. Abdulrahman, R., Neagu, D., Holton, D., Ridley, M., Lan, Y.: Data extraction from online social networks using application programming interface in a multi agent

system approach. In: Nguyen, N. (ed.) Transactions on Computational Collective Intelligence XI, Lecture Notes in Computer Science, vol. 8065, pp. 88–118. Springer Berlin Heidelberg (2013)
2. Agichtein, E., Castillo, C., Donato, D., Gionis, A., Mishne, G.: Finding high-quality content in social media. In: Proceedings of the 2008 International Conference on Web Search and Data Mining. pp. 183–194. WSDM '08, ACM, New York, NY, USA (2008)
3. Bizid, I., Faiz, S., Boursier, P., Yusuf, J.: Integration of heterogeneous spatial databases for disaster management. In: Parsons, J., Chiu, D. (eds.) Advances in Conceptual Modeling, Lecture Notes in Computer Science, vol. 8697, pp. 77–86. Springer (2014)
4. Canali, C., Colajanni, M., Lancellotti, R.: Dataacquisition in social networks: Issues and proposals. In: Proceedings of the International Workshop on Services and Open Sources. SOS'11) (2011)
5. Cappelletti, R., Sastry, N.: Iarank: Ranking users on twitter in near real-time, based on their information amplification potential. In: Proceedings of the 2012 International Conference on Social Informatics. pp. 70–77. SOCIALINFORMATICS '12, IEEE Computer Society, Washington, DC, USA (2012)
6. Imran, M., Castillo, C., Lucas, J., Meier, P., Rogstadius, J.: Coordinating human and machine intelligence to classify microblog communications in crises, pp. 712–721. The Pennsylvania State University (2014)
7. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: Proceedings of the 19th International Conference on World Wide Web. pp. 591–600. WWW '10, ACM, New York, NY, USA (2010)
8. Nakamura, H.: Effects of social participation and the emergence of voluntary social interactions on household power-saving practices in post-disaster kanagawa, japan. Energy Policy 54(0), 397 – 403 (2013), decades of Diesel
9. Pal, A., Counts, S.: Identifying topical authorities in microblogs. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining. pp. 45–54. WSDM '11, ACM, New York, NY, USA (2011)
10. Rieder, B.: Studying facebook via data extraction: The netvizz application. In: Proceedings of the 5th Annual ACM Web Science Conference. pp. 346–355. WebSci '13, ACM, New York, NY, USA (2013)
11. Silva, A., Guimarães, S., Meira, Jr., W., Zaki, M.: Profilerank: Finding relevant content and influential users based on information diffusion. In: Proceedings of the 7th Workshop on Social Network Mining and Analysis. pp. 2:1–2:9. SNAKDD '13, ACM, New York, NY, USA (2013)
12. Smailovic, V., Striga, D., Mamic, D.P., Podobnik, V.: Calculating user's social influence through the smartsocial platform. In: 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM). pp. 383–387 (Sept 2014)
13. Smailovic, V., Striga, D., Podobnik, V.: Advanced user profiles for the smartsocial platform: Reasoning upon multi-source user data. In: Web Proceedings of the 6th ICT Innovations Conference 2014. pp. 258–268 (2014)
14. Weng, J., Lim, E.P., Jiang, J., He, Q.: Twitterrank: Finding topic-sensitive influential twitterers. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining. pp. 261–270. WSDM '10, ACM, New York, NY, USA (2010)
15. Xianlei, S., Chunhong, Z., Yang, J.: Finding domain experts in microblogs. In: Proceedings of the Tenth International Conference on Web Information Systems and Technologies. WEBIST'14 (2014)