Analysis of Tweets to Find the Basis of Popularity

Rajat Kumar Mudgal and Rajdeep Niyogi(✉)

Department of Computer Science and Engineering,
Indian Institute of Technology Roorkee, Roorkee 247667, India
rajatmudgal17@gmail.com, rajdpfec@iitr.ac.in

Abstract. Smart and intelligent recommendation systems can be designed based on analyzing the tweets. Our work is aimed at analyzing the tweets to find the basis of popularity of a person. Although there are some works that have analyzed tweets to detect popular events, not much emphasis has been given to find out the reason behind the popularity of a person based on tweets. In this paper, we suggest an algorithm to find out the reason behind the popularity of a person. We have implemented our algorithm using 2,18,490 tweets of 5 different countries. The results are quite encouraging.

Keywords: Popular person · Twitter user · Popularity · Event detection

1 Introduction

The advent of online social media and its continuous growing popularity has provided a new channel and arena for exchange and/or sharing of information [1, 2]. People on online social media have got an open platform to share opinions, viewpoints, and information on any topic. Over the last few years, Twitter, a micro-blogging service, has gained popularity as one among the most prominent information dissemination and news source agent. Exchange of messages on social media [3] increases considerably with the occurrence of an event that may be related to, for example, social cause, disaster, politics, or a particular person. Users sign in to their Twitter or other social media accounts, to either spread the information or to get updates about the information. Twitter can thus be used to analyze the ongoing situation since it is being used by public and thus it has the potential to provide real-time information. Content on Twitter supplies rich information related to the occurred activity. However, such abundant information is often not trustworthy since it may also contain fake information.

There has been a lot of interest to analyze twitter content [4] which includes, for instance, work in the field of event detection, user selection, and classification of tweets. Besides knowing information about popular people on twitter, it may be useful to know what event has caused the popularity. Such information can let the users know about the arena of popular person and other attributes of the person which can enhance the knowledge of users about famous personalities [5–7]. Moreover, a user may be

O. Gervasi et al. (Eds.): ICCSA 2017, Part I, LNCS 10404, pp. 693–704, 2017.
DOI: 10.1007/978-3-319-62392-4_50

interested to get suggestion about the people she wishes to follow on the basis of the area. The users would like to be kept updated with the currently ongoing events that may lead to the rise or fall of a known figure.

There are multiple sources of information like television, newspapers, social network sites or mouth to mouth words from friends and family [8]. A user may be interested to know about the person who is on everybody's mind in recent times and also wishes to know the reason behind it. To achieve this, initially the current popular persons are obtained from data. In order to find the reasons of popularity, categorization of tweets is carried out since a person may be popular because of more than one reason at a time but there would be one prime reason. By applying all these techniques we can provide better information to the users.

The aim of this paper is to design a method for detecting the popularity of a person and the reason causing the popularity. We use the tweets of different users related to a particular person. We used Twitter4j api in Java to collect the tweets, initially for user selection, and then later to get data about that user. This approach uses nouns in the tweets as their keyword and combines tweets together into a single reason when their match score is above some threshold. Classification of tweets to which category (like business, politics, technology etc.) is realized by categorizing keywords used in each tweet.

The paper is organized as follows. Section 2 describes the related work. Section 3 describes our method for popularity detection. Section 4 describes the implementation details and results obtained by our method. Conclusion and future works are given in Sect. 5.

2 Related Work

A considerable amount of work has been done in classification of tweets, sentiment analysis, and detection of events from tweets. Different approaches have been proposed for sentiment analysis, finding sentiments in words, sentences, topics. Some approaches use natural language processing, some uses pattern based approach and some takes into account machine learning.

In [9], a technique for constructing a Key Graph is suggested using the keywords in the tweets to detect events. This approach is dependent on the interdependency between the keywords. The Key Graph is comprised of nodes and edges where nodes correspond to keywords and the occurrence or the existence of two keywords simultaneously in a tweet is represented by an edge between the nodes. Clusters are created from the Key Graph by clustering different nodes together using a community detection algorithm. In [10], the authors suggest an algorithm called NED (new event detection) to detect events. It consists of two subtasks that are online and retrospective; online NED detects new events in the stream of text while in retrospective NED, unidentified events are detected.

Wavelet transformation is used for event detection in [11]. The problem of identifying events and their user contributed social media documents as a clustering task, where documents have multiple features, associated with domain-specific similarity metrics [12] and pheromone based techniques [13–15]. A general online clustering framework, suitable for the social media domain is proposed in [16]. Several techniques for learning a combination of the feature-specific similarity metrics are given in [16] that are used to indicate social media document similarity in a general clustering framework. In [16] a clustering framework is proposed and the similarity metric learning technique is evaluated on two real-world datasets of social media event content.

Location is considered in [17] with every event as incident location and event are strongly connected. The approach in [17] consists of the following steps. First, preprocessing is performed to remove stop words and irrelevant words. Second, clustering is done to automatically group the messages in the event. Finally, a hotspot detection method is performed.

TwitInfo is a platformfor exploring Tweets regarding to a particular topicis presented in [18]. The user had to enter the keyword for an event and TwitInfo has provided the message frequency, tweet map, related tweets, popular links [19, 20] and the overall sentiment of the event. The*TwitInfo*user interface contained following thing: the user defined name of the event with keywords in the tweet, timeline interface with y axis containing the volume of the tweet, Geo location along with that event is displayed on the map, Current tweets of selected event are colored red if the sentiment of the tweet is negative or blue if the sentiment of the tweet is positive and Aggregate sentiment of currently selected event using pie charts.

TwitterMonitor system is presented in [21] that detect the real time events in defined time window. This is done in three steps. In first step bursty keywords are identified, i.e. keywords that are occurring at a very high rate as compared to others. In second step grouping of bursty keyword is done based on their occurrences. In third and last step additional information about the event is collected.

A news processing system for twitter called as *TwitterStand* is presented in [22]. For users, 2000 handpicked seeders are used for collecting tweets. Seeders are mainly newspaper and television stations because they are supposed to publish news. After that junk is separated from news using the naïve Bayes classifier. Online clustering algorithm called leader-follower clustering to cluster the tweets to form events. A statistical method *MABED* (mention-anomaly-based event detection) is proposed in [23]. The whole process of event detection is divided in three steps. In first step detected the events based on mention anomaly. Second, words are selected that best describes each event. After deleted all the duplicated events or merged the duplicate events. Lastly, a list of top k events is generated.

In [24] a co-relation between clustering and event detection is shown. An aggregate trend change is similar to event detection. To find the popular event, authors of [24] have used algorithms based on community detection. In [26] to find the clusters the authors have suggested a hierarchical clustering of tweets along with the dynamic cutting and rating of resultant clusters is used, a similar technique has been applied in systematic search of maximal length codes [27]. In [28] a technique for finding bursty words is used for detecting events and location recognition using modules.

In [25] it has been stated that an event is associated with the message context but also with the location information, since location is also an important factor of an event. Localized events like any emergency event or any public event, emergency would be more accurately messaged or tweeted by the users closer to the event location in comparison to other users. Hence such users can play the role of sensors – human sensors for briefing an event.

A considerable amount of work has also been carried in the field of sentiment analysis that stresses on finding the sentiments in topics, sentences and the words. Various approaches have been suggested to carry out the sentiment analysis, these approaches either make use of natural language or pattern based processing or machine learning.

In [29] for sentiment analysis authors have suggested a sentiment tree bank approach that is based on a recursive neural network. It calculates in a bottom up manner the parent node vectors and takes advantage of a composition function and also the node vector that features for that node. In [30] an approach has been suggested for finding the sentiment score of informal, short text and also the sentences that consists of phrases within themselves.

Two methods for classification of the Twitter trending topics are proposed in [31] first, based on textual information and the other based on the network structure. In text based model all the hyperlinks are removed from the tweet and then a tokenizer removes stop words and delimited character. Since there is a limitation of 140 characters in a tweet, people use acronyms for words and so a vocabulary is used that has the full form of these words (e.g., BR is used to represent best regard). The network based approach uses a similarity model to find out the trending topic say X. It searches for five topics that are similar to the topic X and finds out the similarity index [5].

Most of the above works are related to sentiments, recommendation systems, trending topic and considered temporal context of messages and classification of tweets. However, these works do not discuss about the rising or decreasing popularity of a person and the reasons behind it. Our approach is different from others as we first look for the popular person and also let the users know the reason behind the popularity.

3 Proposed Methodology

An approach to extract a popular person from tweets is to find a person's name and storing tweet counts corresponding to the person. In order to find the reasons behind the popularity of a person we are using keywords of tweets corresponding to the person.

3.1 Architecture

Figure 1 shows the basic flow diagram of our method. First, we download tweets of different users from different countries and then we look for the person that has been most talked about among those tweets. Then we fetch tweets of that specific person from our database. To detect the reason of popularity we divide all the tweets related to that person into keywords and separate hashtags. Keywords in a tweet are names of things (e.g., name of a person, name of a city). Hashtag is represented using the symbol # followed by some meaningful word like 'Olympics2016'. If two tweets have the same hashtag, it means that these tweets are related and the tweets can be merged into one single tweet.

First we will check hashtag of tweet with events which are already found. Then we pass keywords of that tweet with keywords of events, which are already found into a function called similarity. Similarity we are finding as number of common keywords divided by number of total different keywords. And for every found event with which event, similarity is maximum and greater than threshold then we add tweet into that event. Like this for all tweets algorithm is performed. In the end we find out main reasons behind popularity of person. Then we classify tweets of that person for showing the interest of users towards that popular person means what general users think about that person. Here user is the twitter user, whose tweets are downloaded from twitter.

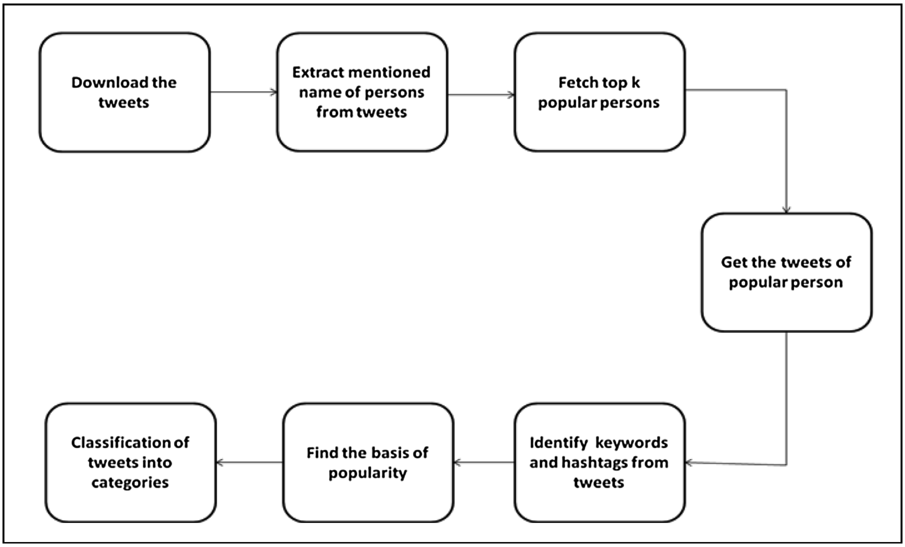

Download the tweets
Extract mentioned name of persons from tweets
Fetch top k popular persons
Get the tweets of popular person
Identify keywords and hashtags from tweets
Find the basis of popularity
Classification of tweets into categories

Fig. 1. Overview of the proposed method

3.2 Data Collection

We collected 2,18,490 tweets of 5 different countries from September, 2016 to November, 2016 using Twitter4j API [33]. Tweets were downloaded by taking latitude and longitude values of countries. We took news channels (CanadaNews, bbcnews) into consideration because news channels are reliable sources of data; news channels produce more data than simple twitter users.

3.3 Extraction of Names of Persons from Tweet

We used Stanford Named Entity Recognition (NER) tagger [32] for extracting the names of persons from tweets. NER labels sequence of words into a text which contains names of things, such as name of person, name of company, and name of place. Every tweet is passed through the NER tagger and it returns names of things for every tweet. We store only names of persons, and for this we used a hash function.

3.4 Fetching Top k Popular Persons

For storing name of a person and the number of occurrences of the names, we use a hash table named h_table, that has two fields: key and value. In the key field, we store person name; in the value field, a tuple <tweet_id, count_name>. If a name of a person does not exist, the count of the person is set to 1 and add the corresponding tweet id. Otherwise, increment count by one and update tweet id field.

3.5 Find the Basis of Popularity

For storing hashtags and keywords and the corresponding tweet ids and count of reasons of popularity, we use hash table named H_table, that has two fields key and value. In the key field, we store a tuple <hashtag, keywords>; in the value field, a tuple <tweet_id, count_reason>.

The following symbols are used in the algorithm.

S: set of all tweets storing tweet ids along with person mentioned in tweet.
R: set of all reasons related to popular person. Initially this set is empty.
PT: set of all the tweets of popular persons along with its keywords and hashtags.
h_table: a hash table that is initially empty.
H_table: a hash table that is initially empty.
P: set of popular persons.
m: threshold value, $0 < m < 1$.

Algorithm for discovering reason of popularity

```
Input: S        output: R
Step 1: Get the tweet count for each person
    for each tweet t ∈ S do
        for each name ∈ t do
            if name exists in h_table then
                add tweet_id in h_table[name][0];
                increment the count field of h_table[name][1] by1;
            else
                create new entry h_table[name] ;
                add tweet_id in h_table[name][0];
                set h_table[name][1] to 1;
        end for
    end for
Step 2: Sort h_table on the basis of count in descending order. Find the top k
        popular persons and store them into P.
Step 3: Get all the tweets of P from S and break down these tweets into keywords
        and hashtags and store them into PT.
Step 4: Get all the reasons related to first k popular persons.
    for each person p ∈ P
        for each tweet t of p in PT
            tag := hashtag of t; kw := keyword of t;
            flag := false;
            for each key k of H_table
                if tag = k[0] then
                    flag := true;
                    k[1] := k[1] ∪ kw;
                    increment H_table[k][1] by 1;
                else if similarity (kw, k[1]) > m then
                    flag := true;
                    k[1] := k[1] ∪ kw;
                    increment H_table[k][1] by 1;
            end for
            if flag = false then
                add tweet id into H_table[(tag, kw)][0]
                set H_table[(tag, kw)][1] to 1;
        end for
    end for
Step 5: sort H_table according to field of tweet count in descending order and store
into R. Find the top popular reasons r from the set R having maximum tweet count.
```

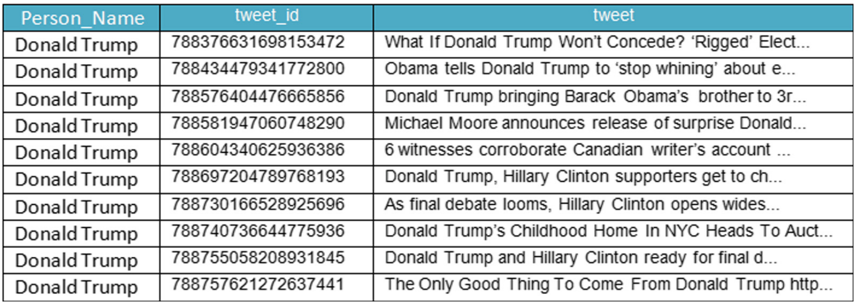
4 Implementation and Results

To implement the algorithm, we collected 2,18,490 tweets of 5 different countries, using Twitter API. First, a user provides the value of n i.e., top n popular persons according to the downloaded tweets. Table 1 shows the output when a user provides the value of n = 4.

Table 1. Top n (n = 4) popular persons and their tweet count

Sl. no.	Person_name	Tweet_count
1.	Donald Trump	13117
2.	Hillary Clinton	9934
3.	Justin Trudeau	5432
4.	Malcolm Turnbull	5048

Once the user gets the top n popular persons, she can select any one person from the results to get more details of the selected person. In this interface, on selecting one person it will show all the tweets of that person. User can get more information about the person using these tweets. Figure 2 shows the output of selecting one person.



Person_Name	tweet_id	tweet
Donald Trump	788376631698153472	What If Donald Trump Won't Concede? 'Rigged' Elect...
Donald Trump	788434479341772800	Obama tells Donald Trump to 'stop whining' about e...
Donald Trump	788576404476665856	Donald Trump bringing Barack Obama's brother to 3r...
Donald Trump	788581947060748290	Michael Moore announces release of surprise Donald...
Donald Trump	788604340625936386	6 witnesses corroborate Canadian writer's account ...
Donald Trump	788697204789768193	Donald Trump, Hillary Clinton supporters get to ch...
Donald Trump	788730166528925696	As final debate looms, Hillary Clinton opens wides...
Donald Trump	788740736644775936	Donald Trump's Childhood Home In NYC Heads To Auct...
Donald Trump	788755058208931845	Donald Trump and Hillary Clinton ready for final d...
Donald Trump	788757621272637441	The Only Good Thing To Come From Donald Trump http...

Fig. 2. Tweets corresponding to the selected popular person

For the selected person, the reasons of popularity are given in Table 2. The table lists person name, all the popularity reasons, and the corresponding tweet counts.

Table 2. Reasons of popularity of the selected person

Person_name	Popularity_reason	Tweet_count
Donald Trump	Election2016	1387
Donald Trump	Campaign	948

The pie chart in Fig. 3 shows users' interest towards the selected popular person (Donald Trump). Since a large percentage of tweets are related to politics, this indicates that users are showing interest in political aspects of the person.

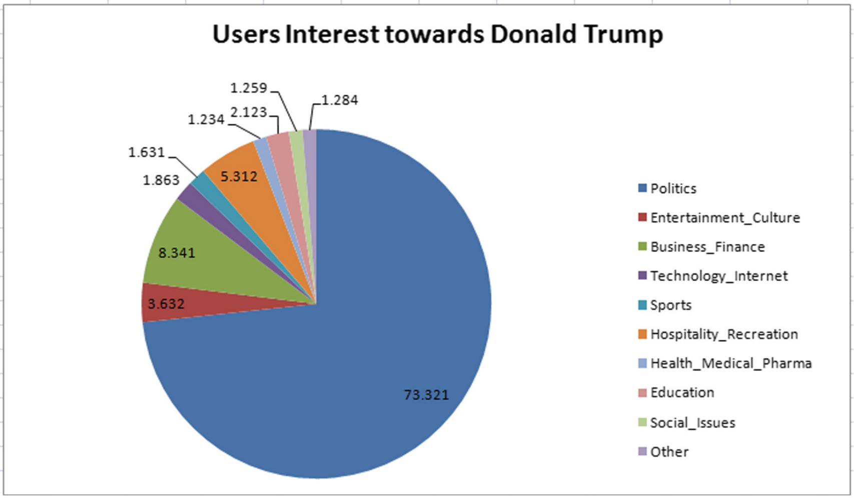


Fig. 3. Pie chart representing classification of tweets according to users' interest toward popular person Donald Trump

We can compare users' views for two different popular persons. Figure 4 shows users' views for Donald Trump and Malcolm Turnbull. From this Figure we can conclude that in politics, users are more interested toward Trump than Turnbull.

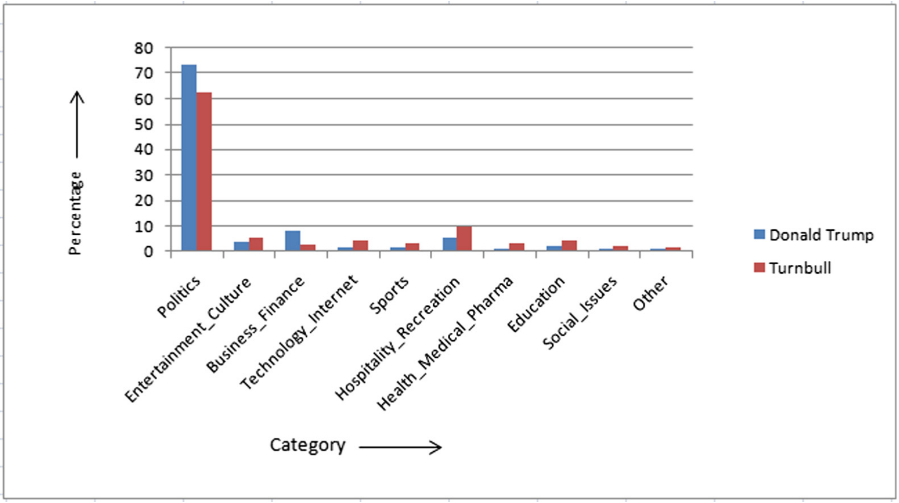


Fig. 4. Comparison between the tweets related to Trump and Turnbull

5 Conclusion and Future Work

In this paper, we suggested an approach to get the popular person from the gathered tweets and obtained the reason behind the popularity of that person. In our approach, we first look for the names mentioned in the tweets and the name that occurs with highest frequency is suggested as the most popular person. In order to find the reason behind the popularity of the person we developed an algorithm that looks for the possible events in the tweets. For implementation, we used data sets of different time frames to showcase the output and the results obtained are very encouraging. In future we would like to further extend our system to compare the top most popular persons with each other and look if they are inter connected by the same reason or not.

References

1. Franzoni, V., Mencacci, M., Mengoni, P., Milani, A.: Heuristics for semantic path search in Wikipedia. In: Murgante, B., et al. (eds.) ICCSA 2014. LNCS, vol. 8584, pp. 327–340. Springer, Cham (2014). doi:10.1007/978-3-319-09153-2_25
2. Franzoni, V., Milani, A.: Semantic context extraction from collaborative networks. In: Proceedings of the 2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2015, pp. 131–136. IEEE Press (2015)
3. Leung, C.H.C., Chan, A.W.S., Milani, A., Liu, J., Li, Y.: Intelligent social media indexing and sharing using an adaptive indexing search engine. ACM Trans. Intell. Syst. Technol. **3**(3), 221–238 (2012). ACM Press
4. Leung, C.H.C., Li, Y., Milani, A., Franzoni, V.: Collective evolutionary concept distance based query expansion for effective web document retrieval. In: Murgante, B., Misra, S., Carlini, M., Torre, C.M., Nguyen, H.-Q., Taniar, D., Apduhan, B.O., Gervasi, O. (eds.) ICCSA 2013. LNCS, vol. 7974, pp. 657–672. Springer, Heidelberg (2013). doi:10.1007/978-3-642-39649-6_47
5. Franzoni, V., Milani, A.: PMING distance: a collaborative semantic proximity measure. In: Proceedings - 2012 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2012, vol. 2, pp. 442–449. IEEE Press (2012)
6. Milani, A., Santucci, V.: Community of scientist optimization: an autonomy oriented approach to distributed optimization. AI Commun. **25**(2), 157–172 (2012). IOS Press
7. Franzoni, V., Leung, C.H.C., Li, Y., Mengoni, P., Milani, A.: Set similarity measures for images based on collective knowledge. In: Gervasi, O., Murgante, B., Misra, S., Gavrilova, M.L., Rocha, A.M.A.C., Torre, C., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2015. LNCS, vol. 9155, pp. 408–417. Springer, Cham (2015). doi:10.1007/978-3-319-21404-7_30
8. Franzoni, V., Mencacci, M., Mengoni, P., Milani, A.: Semantic heuristic search in collaborative networks: measures and contexts. In: Proceedings - 2014 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2014, pp. 187–217. IEEE Press (2014)
9. Sayyadi, H., Hurst, M., Maykov, A.: Event detection and tracking in social streams. In: Proceedings of the Third International ICWSM Conference, pp. 311–314 (2009)
10. Dou, W., Wang, X., Ribarsky, W., Zhou, M.: Event detection in social media data. In: Proceedings of IEEE VisWeek Workshop on Interactive Visual Text Analytics-Task Driven Analytics of Social Media Content, pp. 971–980 (2012)

11. Weng, J., Yao, Y., Leonardi, E., Lee, F.: Event detection in Twitter. In: Proceedings of the 5th International AAAI Conference on Weblogs and Social Media, pp. 401–408 (2011)
12. Becker, H., Naaman, M., Gravano, L.: Learning similarity metrics for event identification in social media. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 291–300 (2010)
13. Baioletti, M., Milani, A., Poggioni, V., Rossi, F.: Experimental evaluation of pheromone models in ACOPlan. Ann. Math. Artif. Intell. **62**(3–4), 187–217 (2011). Springer
14. Ukey, N., Niyogi, R., Singh, K., Milani, A., Poggioni, V.: A bidirectional heuristic search for web service composition with costs. Int. J. Web Grid Serv. **6**(2), 160–175 (2010). Inderscience
15. Milani, A., Poggioni, V.: Planning in reactive environments. Comput. Intell. **23**(4), 439–463 (2007). Wiley
16. Becker, H., Naaman, M., Gravano, L.: Beyond trending topics: real-world event identification on Twitter. In: Proceeding of Fifth International AAAI Conference on Weblogs and Social Media (2011)
17. Unankard, S., Li, X., Sharaf, M.A.: Emerging event detection in social networks with location sensitivity. In: Proceedings of World Wide Web, pp. 1–25 (2014)
18. Marcus, A., Bernstein, M.S., Badar, O., Karger, D.R., Madden, S., Miller, R.C.: TwitInfo: aggregating and visualizing microblogs for event exploration. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 227–236 (2011)
19. Chiancone, A., Franzoni, V., Niyogi, R., Milani, A.: Improving link ranking quality by quasi-common neighbourhood. In: Proceedings - 15th International Conference on Computational Science and Its Applications, ICCSA 2015, pp. 21–26. IEEE Press (2015)
20. Franzoni, V., Milani, A.: Heuristic semantic walk. In: Murgante, B., Misra, S., Carlini, M., Torre, C.M., Nguyen, H.-Q., Taniar, D., Apduhan, B.O., Gervasi, O. (eds.) ICCSA 2013. LNCS, vol. 7974, pp. 643–656. Springer, Heidelberg (2013). doi:10.1007/978-3-642-39649-6_46
21. Mathioudakis, M., Loudas, N.: TwitterMonitor: trend detection over the twitter stream. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, pp. 1155–1158 (2010)
22. Sankaranarayanan, J., Samet, H., Teitler, B.E., Lieberman, M.D., Sperling, J.: TwitterStand: news in Tweets. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 42–51 (2009)
23. Guille, A., Favre, C.: Event detection, tracking, and visualization in twitter: a mention anomaly based-approach. Proc. Soc. Netw. Anal. Min. **5**(1), 1–18 (2015)
24. Aggarwal, C.C., Subbian, K.: Event detection in social streams. In: Proceeding of SDM, vol. 12, pp. 624–635 (2012)
25. Abdelhaq, H., Sengstock, C., Gertxz, M.: EvenTweet: online localized event detection from twitter. Proc. VLDB Endow. **6**(12), 1326–1329 (2013)
26. Ifrim, G., Shi, B., Brigadir, I.: Event detection in Twitter using aggressive filtering and hierarchical Tweet clustering. In: Proceedings of SNOW WWW Workshop (2014)
27. Marcugini, S., Milani, A., Pambianco, F.: NMDS codes of maximal length over Fq, $8 \leq q \leq 11$. IEEE Trans. Inf. Theory **48**(4), 963–966 (2002). IEEE Press
28. Wang, X., Zhu, F., Jiang, J., Li, S.: Real time event detection in Twitter. In: Wang, J., Xiong, H., Ishikawa, Y., Xu, J., Zhou, J. (eds.) WAIM 2013. LNCS, vol. 7923, pp. 502–513. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38562-9_51
29. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642. Citeseer (2013)

30. Kiritchenko, S., Zhu, X., Mohammad, S.M.: Sentiment analysis of short informal texts. J. Artif. Intell. Res. **50**(1), 723–762 (2014)
31. Lee, K., Palsetia, D., Narayanan, R., Patwary, M.M.A., Agrawal, A., Choudhary, A.: Twitter trending topic classification. In: 11th IEEE International Conference on Data Mining Workshops, pp. 251–258, December 2011
32. Finkel, J., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics ACL 2005, pp. 363–370 (2005)
33. Twitter: Twitter Developers. https://dev.twitter.com