Dynamic Topic-Related Tweet Retrieval

Juan M. Cotelo, Fermin L. Cruz, and Jose A. Troyano
Department of Languages and Computer Systems, University of Seville, 41012 Seville, Spain. E-mail: {jcotelo, fcruz, troyano}@us.es

Twitter is a social network in which people publish publicly accessible brief, instant messages. With its exponential growth and the public nature and transversality of its contents, more researchers are using Twitter as a source of data for multiple purposes. In this context, the ability to retrieve those messages (tweets) related to a certain topic becomes critical. In this work, we define the topic-related tweet retrieval task and propose a dynamic, graph-based method with which to address it. We have applied our method to capture a data set containing tweets related to the participation of the Spanish team in the Euro 2012 soccer competition, measuring the precision and recall against other simple but commonly used approaches. The results demonstrate the effectiveness of our method, which significantly increases coverage of the chosen topic and is able to capture related but unknown à priori subtopics.

Introduction

Recently, Twitter has received more attention from the scientific community, mostly because of its high potential for opinion analysis, hot topic retrieval, and the relative success of Twitter attributable to recent widespread smartphone usage. However, Twitter is a vast social network in which searching and retrieving topic-specific tweets are not easy tasks. Although it provides a keyword-based interface similar to any modern search engine, it is limited in scope and usefulness, being more a simple tool for the common user than for serious data extraction. Furthermore, devising an effective keyword set is a difficult task because of the following two main factors: the difficulty of establishing an adequate keyword set representing a topic in advance and the real-time changing nature of Twitter.

We think that, before trying to analyze topic-specific data retrieved from Twitter, the data retrieval process has to be addressed in a systematic way, ensuring some degree of quality and significance of the extracted data. This idea led us to a task not previously defined: the retrieval of all tweets related to a given topic during a specific time window. A formalization of the task is necessary to identify the main issues of the task and to make it easier to devise a systematic method to address this task. Thus we present a formalization of this task and provide a general method for addressing it using the underlying graph structure of Twitter.

Most research work that makes use of data from Twitter relies on querying for data retrieval, and, depending on the problem addressed in each scenario, this may result in significant loss of data or data noise. This is because, in recent work on Twitter, data retrieval is not the main concern. Most of the work provides only lists of terms selected manually, mainly used to build queries for tweet retrieval.

Previous work, such as Tumasjan, Sprenger, Sandner, and Welpe (2010); Gayo-Avello, Metaxas, and Mustafaraj (2011); Hong and Nadler (2011); Pennacchiotti and Popescu (2011); Congosto, Fernández, and Egido (2011); Agarwal, Xie, Vovsha, Rambow, and Passonneau (2011); Davidov, Tsur, and Rappoport (2010a, 2010b); Go, Bhayani, and Huang (2009); Jiang, Yu, Zhou, Liu, and Zhao (2011); Kim, Gilbert, Edwards, and Graeff (2009); Pak and Paroubek (2010); Silva, Gomide, Veloso, Meira, and Ferreira (2011); and Tan et al. (2011), make use only of manually crafted lists of terms matching their specific needs, usually selecting users and hashtags based on criteria related to the topics. This is not a general method, and data coverage may be insufficient for the specific task.

As an exception, a custom methodology for tweet retrieval has been presented by Golbeck and Hansen (2011), starting from a seed set made of congressmen Twitter users and expanding it through its followers, trying to establish some link between congressmen and communication media using followers as an intermediate link. However, this method is neither general nor dynamic, and is inappropriate for other scenarios.

An effort worth mentioning, which comes from the information retrieval (IR) scientific community, is the 2012 edition of the TREC Microblog track (Soboroff, Ounis, & Lin, 2012). This track shares some similarities with the main goal of our work, but its focus is clearly distinct from ours; the main search task presented in that track is a real-time ad hoc task, consisting of retrieving the most recent but relevant information to a provided query. Hence, the goal is answering a specific query, taking into account the time of the query, returning relevant tweets from newest to oldest. Those lookups are made against a previously collected corpus, designed by the NIST.

The absence of approaches that systematically address the data retrieval process, ensuring some degree of quality and significance, is one of the driving forces for making this effort. Our contributions can be summarized as follows:

- We introduce the problem of topic-related tweet retrieval, showing some peculiarities to be considered when using data from Twitter.
- We provide a formalization of the topic-related tweet retrieval and identifying subtasks within it.
- We propose a general method for addressing the task, exploiting the underlying structure of the Twitter network.
- We perform an experimentation step in which we use our method and compare it against other typical approaches, demonstrating its effectiveness.
- We perform an analysis of the retrieved data, addressing some interesting factors regarding the method and proposing further guidelines to optimize the method.

The paper is organized as follows. In the Task Definition section, we introduce a problem and provide a formalization of the task addressing it. In the Proposed Method section, we propose the above-mentioned method addressing the task. In Experimentation, we describe the experimentation step performed and present a comparison against the typical approaches taken in the above-mentioned work. In Analysis of Results, we perform a more detailed analysis on the data gathered by the method, describing some important considerations. Finally, we provide a Conclusions and Future Work section.

Task Definition

The goal of this task is the retrieval of all tweets related to a given topic that were sent during a given time window. To understand the complexity of the task, we clarify three major concepts that define the whole task: Twitter, topic, and time. User messages in Twitter (or tweets) have some special characteristics to be considered.

First, there is a restriction regarding the length of the tweets, 140 characters being the upper limit, something that is not unusual, inasmuch as short messages are part of any microblogging network. This restriction usually leads to shortened words, a plethora of acronyms, and similar techniques to overcome the imposed limit, lowering the quality of the text overall. URL shorteners have become a necessity when including hyperlinks in the text.

Second, the content of the tweets is in plain text, and Twitter provides very few special constructs (also written in plain text) for composing tweets, only allowing direct mentioning of other users (user names prefixed with the character "@") and hashtags. These hashtags are words or even phrases prefixed with the character "#" within the composed message and are used mainly for unmoderated ad-hoc discussion forums associated with some specific topic, being, in fact, metadata tags but completely unmanaged by the platform. Furthermore, hashtags are promoted by individuals and sometimes are used as "beacons" or "flags," grouping loosely related tweets under that hashtag.

Third, the only way to retrieve tweets from Twitter is by making direct queries composed of keywords (words, tags, or even usernames) resembling typical search engine interfaces such as Google. This greatly restricts any approach designed for solving the topic-related retrieval task.

The concept of topic is easy to understand intuitively but hard to define. Broadly speaking, we can define it as the main subject of a discussion between users. Furthermore, the realization of the desired topic may be difficult by itself, finding issues in "transferring" the abstract concept to something that may be consumed by systems. With regard to time, another factor to be considered is the very temporal nature of Twitter. Being a social network that changes in real time with fast responses from the community, new trends and hashtags are being created constantly, making the task of tracking topics over time more difficult.

These factors render the retrieval of tweets related to a topic an interesting, nontrivial task: composing queries that ensure the retrieval of tweets related along the timeline. After this intuitive explanation of the task, we provide a formalization of it (definition 1) as a starting point for the rest of the article.

Definition 1: Given the following provisions:

- Let T be the set of all existing tweets in Twitter. T continuously grows over time, so we denote as T^t the set of all existing tweets in a t instant.
- Let Q be the set of all queries that may be made on T. Each query, q, is a specific set of keywords whose execution in an instant of time t returns $T_q^t \subseteq T^t$; that is, a set of tweets of T^t that have any of the keywords of q.
- Let $T_{topic}^t \subseteq T^t$ be the set of all existing tweets in Twitter in the t instant that are related to some specific topic.
- Let $Q_{topic}^t \subseteq Q$ such that $Q_{topic}^t = \{q_i : T_{q_i}^t \supseteq T_{topic}^t\}$.

The topic-related tweet retrieval task, given a t instant of time, consists of:

1. Characterizing a $q_i \in Q_{topic}^t$, preferably with a small $|T_{q_i}^t|$, limiting the number of retrieved tweets not related to the topic.

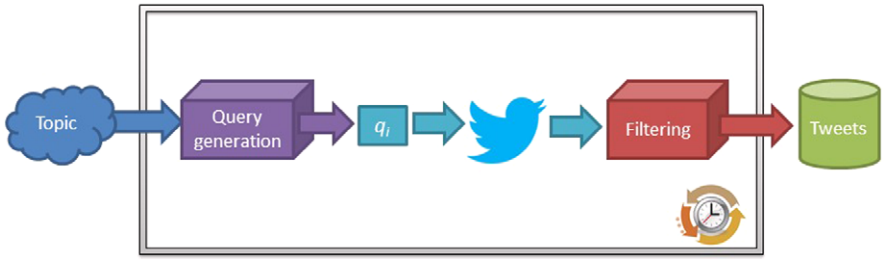

FIG. 1. Topic-related tweet retrieval task. [Color figure can be viewed in the online issue,which is availableat wileyonlinelibrary.com.]

2. Selecting the tweets from the resulting $T^t_{q_i}$ belonging to T^t_{topic}, removing the retrieved tweets not related to the topic.

This formalization naturally breaks down the task into two subtasks, each one related to points 1 and 2 of definition 1, respectively: query generation and filtering. Although the two tasks are related, this division allows us to focus on each subtask independently, and solving the challenges of each becomes easier and more efficient.

Furthermore, each subtask has different roles within the system, altering the performance of the system in different ways. The query generation subtask greatly affects the recall of the whole system because the retrieved tweets in the first instance depend on the generated query. Generating broad queries may increase the recall but will introduce much noise. On the other hand, the filtering subtask does a posteriorly directed selection of the retrieved tweets, minimizing noise and largely affecting the precision of the whole system, but it may discard some tweets that are related to the topic.

Figure 1 shows the main elements of the task mentioned above. The "time" concept is represented by the clock image in the diagram. The entire task not only is hard to characterize but also hard to solve and evaluate. The technical peculiarities of Twitter, the difficulty of realizing the desired topic in something less abstract, the very nature of the users and the messages composed by them, and the temporal behavior of the task are the most prominent issues. As we mentioned above, this task is not at all trivial.

Proposed Method

In this section, we address the topic-related retrieval task, addressing exclusively the query generation subtask. Furthermore, as mentioned in the previous section, the query generation subtask greatly affects coverage, so the main focus of this method is increasing coverage without incurring an unacceptable loss of precision.

Given that Twitter actually is a huge graph with several kinds of relationships between its elements, it is straightforward to take a graph-based approach for representing and analyzing its data. In essence, we build a graph from the tweets using hashtags and users as nodes, perform a relevance ranking on the nodes, select the best nodes, and compose a query with them, with all of this process performed in a iterative way.

Graph Construction

Our idea consists of building a graph from retrieved tweets, generating nodes and edges using the main elements found in tweets: user and hashtags. We find different types of relationships:

- Mentions: the author of a tweet includes the username of any user in the contents of that tweet. This relationship between users often occurs when someone wants to refer to another user or reply to tweets.
- Retweets: the contents of this tweet are essentially the same as any other tweet, prefixed with some kind of construct such as "RT @username:" mentioning the original author or using a retweet button. This is a specific case of the mention relationship, but it replicates the same relationships found in the original tweet because the contents are very similar.
- Simple tag use: the author of a tweet includes a hashtag in the contents of that tweet, thus relating users and hashtags.
- Tag co-occurrence: the author of a tweet includes more than a hashtag in the contents of that tweet. This case is an extension of the previous one, adding a new relationship between two or more hashtags.

From these observed relationships, we build a structural graph representing the current topology between users and hashtags. In this constructed graph, the nodes are the hashtags and the users, and the arcs are the relationships found in the content of the tweets retrieved.

Mentions and retweets are represented via directed arcs from users to other users; simple tag uses are represented via directed arcs from users to tags. If any arc already exists, its weight is incremented, reinforcing that relationship. Notice that a tweet may expose more than one relationship instance, so many elements may be generated from the content of a single tweet, such as multiple mentions or several tags used.

To clarify the graph-building process, a step-by-step toy-grade example is shown below, using the tweet sequence shown in Figure 2 and the breakdown indicating the relationships exposed in that tweets in Table 1.

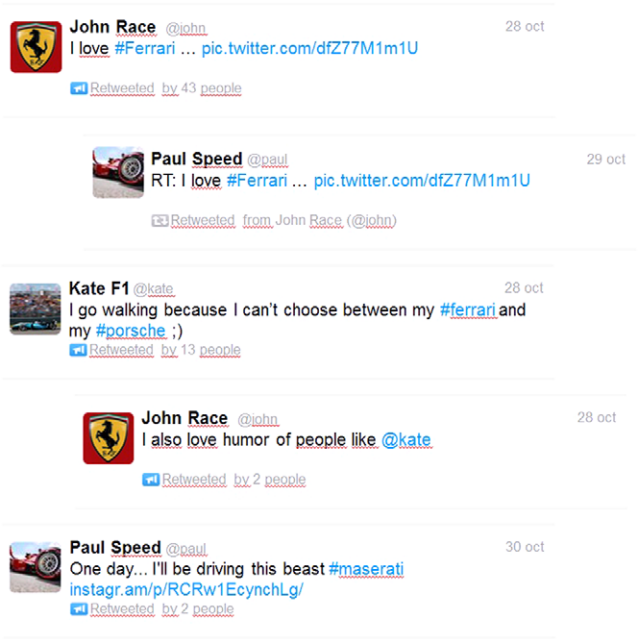

FIG. 2. Sample of an interaction in Twitter related to Ferrari. [Color figure can be viewed in the online issue,which is availableat wileyonlinelibrary.com.]

TABLE 1. Breakdown of an interaction in Twitter related to Ferrari from Figure 2.

Tweet	Author	Relationships	Contents
1	@john	Tag use	I love #Ferrari . . . pic.twitter. com/dfZ77M1m1U
2	@paul	Retweet	RT: I love #Ferrari . . . pic.twitter. com/dfZ77M1m1U
3	@kate	Tag use	I go walking because I can't choose between my #ferrari and my #porsche ;)
4	@john	Mention	I also love humour of people like @kate
5	@paul	Tag use	One day . . . I'll be driving this beast #maserati instagr.am/p/RCRw1EcynchLg/

The whole graph-building process is also shown in Figure 3, with each subfigure being a part of the sequence of the process. The process is performed as follows:

Tweet 1: Create the nodes @john and #ferrari and establish a directed arc from @john to #ferrari.
Tweet 2: Create the node @paul, establish a directed arc from @paul to @john and increment the weight of the arc from @john to #ferrari.
Tweet 3: Create the node @kate, establish a directed arc from @kate to #ferrari and from @kate to #porsche.
Tweet 4: Establish a directed arc from @john to @kate.
Tweet 5: Create the node #maserati and establish a directed arc from @paul to #maserati.

In this example, the resulting graph representing data from five tweets has six nodes and seven arcs with a low density of $D = 0.233$. A real-world example may have millions of nodes and more than tens of millions of arcs and may be denser.

Graph Analysis

After the graph is built, a relevance ranking algorithm is applied to it, obtaining the most relevant nodes in the graph, taking into account the topology of the graph. The algorithm applied to the graph is the well-known PageRank.

PageRank (Page, Brin, Motwani, & Winograd, 1999) is a relevance ranking algorithm originally devised to measure the importance of any Internet web page according to the links that page receives but is general enough for application in contexts other than web pages. Given a graph $G = (V,E)$, where V is a set of vertices and E a set of directed arcs between two vertices, two operations, $In(V_i)$ and $Out(V_i)$, are defined first of all to calculate, respectively, the set of nodes with arcs entering or leaving the vertex V_i. From these two basic operations, we define the score (or PageRank) of a given vertex with the following formula:

$$PR(V_i) = (1-d) + d \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} PR(V_j), \qquad (1)$$

where d is a damping factor that allows the convergence of the algorithm. In the context of navigation on the Internet, this factor represents the probability that a user accesses a page through a link at the current page, making $(1 - d)$ the probability of that user jumping to a random page not linked to the current page. In the original definition of PageRank a value of 0.85 for factor d is recommended.

Starting from arbitrary values for the scores of the nodes of a graph, a convergence point is reached, applying the formula iteratively until the largest difference in the scores obtained for each node, between two iterations, is below a certain threshold. Once the algorithm has finished, the score attained by each node represents the importance thereof and may be used as a criterion for making decisions.

Method Description

The graph construction and analysis are critical parts of our method, but we have to define how exactly to compose the query and which data are used during the graph building. We proposed a cyclic method in which each step in a graph analysis approach is made from previously collected data in order to compose an appropriate query for the next step.

Figure 4 shows the entire process as a specialization of the definition of the task presented in Figure 1, and the method is explained in Algorithm 1. Notice that this method is entirely focused on the query generation subtask and that no filtering step is made.

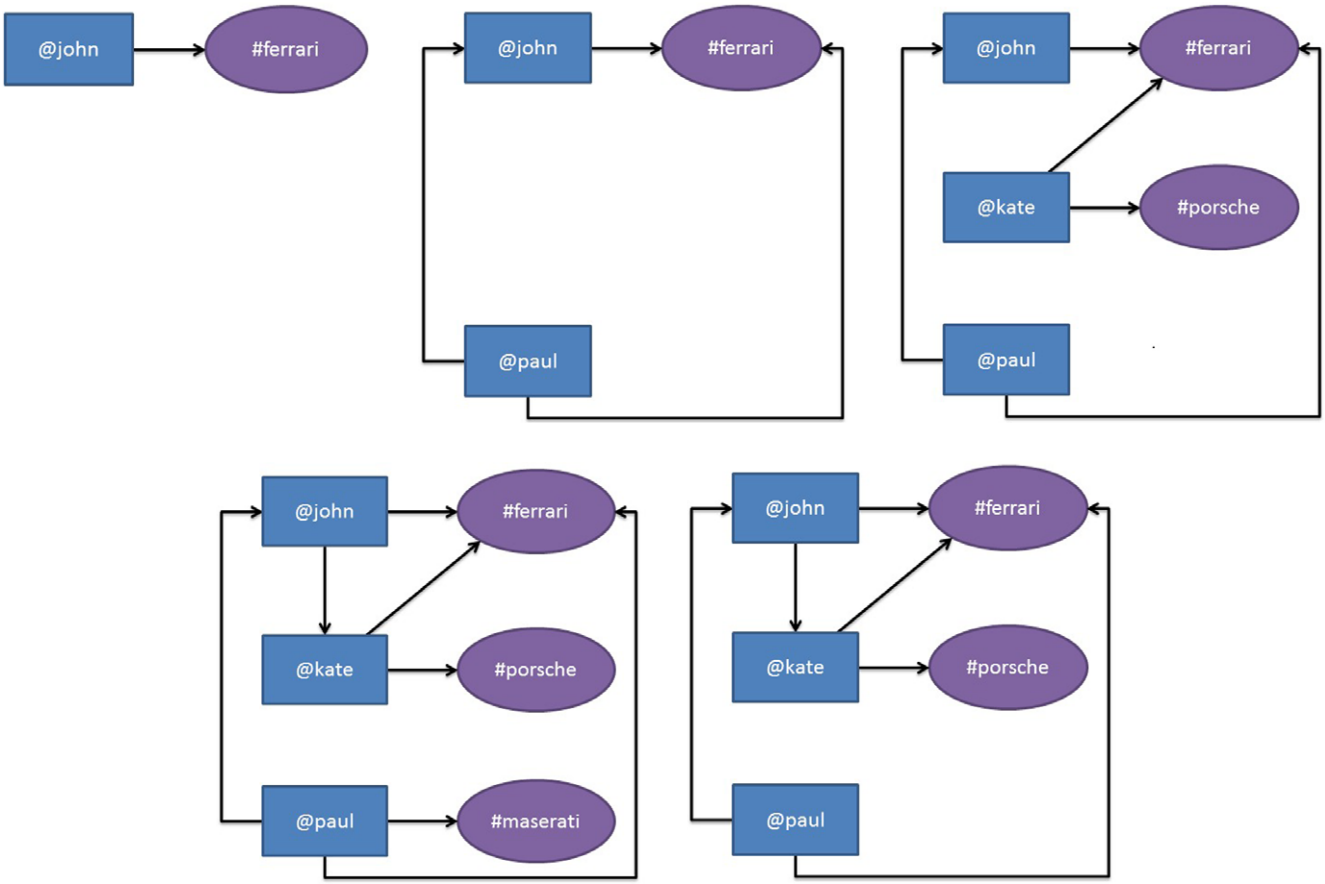

FIG. 3. Step-by-step graph-building process for the sample shown in Figure 2. [Color figure can be viewed in the online issue,which is availableat wileyonlinelibrary.com.]

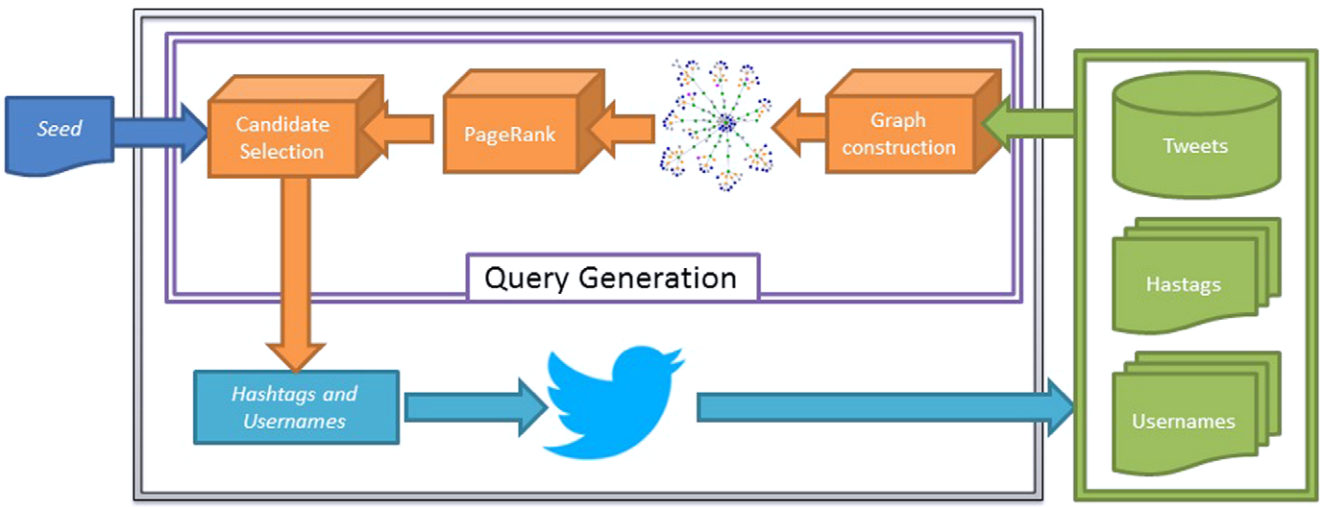

FIG. 4. Proposed method for the topic-related tweet retrieval task. [Color figure can be viewed in the online issue,which is availableat wileyonlinelibrary.com.]

ALGORITHM 1. Proposed Method.

```
Data: Seed term s, analysis window w, retrieval iteration ime i
begin
   Terms ← {s}
   T ← ∅
   repeat
    T ← T ∪ QueryTwiner (Terms, i)
    G ← CreateEmptyGraph()
    Tw ← tweets from T crawled in last w iterations
    foreach tweet t ∈ Tw such term s appears in t do
     user ← user that composed t
     tags ← tags that appear in t
     mentions ← user mentions that appear in t
     if user ∉ G then update the graph with a node representing user

     foreach tag ∉ tags do
      if tag ∉ G then update G with a node representing tag

      if directed are a = (user, tag ) ∉ G then add directed are a to G

      increment the weight of the are a = (user, tag)
     end
     foreach mentioned_user ∉ mentions do
      if mentioned_user ∉ G then update G with a node representing that user

      if directed are a = (user, mentioned_user) ∉ G then add directed are a to G

      increment the weight of the are a = (user, mentioned_user)
    end
   end
   R ← PageRank (G)
   Terms ← first n ranked terms from R
  until end of crawling session
end
```

For every stipulated period, the graph building and analysis process is made from tweets previously collected during a pre-determined time window. This period may span from several minutes to days, but we found that a 60-minute time window was adequate.

The first iteration of the cycle needs some starting keyword set or seed, so the most appropriate seed usually is a previously selected set of keywords representing a central concept of the selected topic. Devising methods for deciding the initial seed is beyond the scope of this paper.

After performing the graph-building step with the recently retrieved data, we use PageRank to make a relevance ranking on the nodes of the graph, which is used as a candidate list. The candidate selection process is made by retaining the best nodes from the ranking and discarding the rest, establishing a threshold point in the list determined beforehand. In this case, we determined that point using an integer parameter, k, representing the maximum size of the set containing the selected elements, fixed ahead of time.

Finally, composing the query is quite straightforward. The selected elements are used as keywords, being concatenated using the typical *or* operation, thus composing the query to be used on Twitter. In essence, we use the same structural constructs found in Twitter (users and hashtags) for the query construction but taking into account the underlying graph.

It is worth mentioning that the proposed method has some resemblance to the IR pseudorelevance feedback model and that the two share common traits. However, our method greatly differs from the pseudorelevance feedback model in some key points: graph construction and analysis instead using a vector space model, taking into account structural elements of tweets instead of their contents, and query generation using those structural elements. Further information may be found in pseudorelevance work (Cao, Nie, Gao, & Robertson, 2008; Lavrenko & Croft, 2001; Lee, Croft, & Allan, 2008; Tao & Zhai, 2006).

Experimentation

To test the performance of the proposed method, we applied it to a concrete topic, performed some analysis on the data set obtained, and made a comparison with simpler approaches and different parameter configurations.

Experimental Setup

The topic selected for the data collection was the 2012 UEFA European Football Championship, also commonly

TABLE 2. Computed precision and interagreement for each reviewer.

	Reviewer 1	Reviewer 2	Reviewer 3
Precision	0.8882	0.8292	0.8715
Agreement with reviewer 1	1.0	0.9290	0.9651
Agreement with reviewer 2	0.9290	1.0	0.9375
Agreement with reviewer 3	0.9651	0.9375	1.0
Complete agreement		0.9158	
Fleiss's kappa		0.7627	

TABLE 3. Precision and data set recall comparison between methods.

Method	Precision	Data set recall
Baseline (central term only)	0.9726	0.1504
Ad hoc static selection	0.9721	0.2698
Method with $k = 10$	0.9274	0.8034
Method with $k = 20$	0.8713	1.0000

referred as Euro 2012, an important pan-European football competition highly tracked in the sports media. We wanted to focus on Twitter messages from a Spanish perspective, so the selected central term as seed for the method was #vamosespaña. This hashtag had a strong reception within the Spanish community, being a common term for cheering up the Spanish football selection.

After selecting the central keyword for the topic, we collected tweets using three different methods throughout the event, spanning from June 6, 2012, to July 3, 2012. The methods we used for the experimentation stage were the following:

- Central keyword: Using the central keyword as the sole query term throughout the event is a naïve approach used as baseline for our experiments with the central keyword #vamosespaña, as mentioned above.
- Static ad hoc list of keywords: This straightforward approach consists of using an expert-made collection of keywords (usually hashtags) as the static query used throughout the event. In this setup, we used the following keyword set: #vamosespaña, #nohaydossintres, #eurocopa, #eurocopa2012, #laroja. These terms were selected for their high relevance and low ambiguity regarding the event.
- Proposed dynamic method: We use the central keyword #vamosespaña as starting seed for the proposed method, varying the keyword set size up to $k = 20$.

Sampling and Tagging Process

We decided to evaluate a significant sample with a significance level of $\alpha = .05$ and an error bound of 1%, resulting in a data set with 10,000 messages, though only 9,604 messages are strictly required. This sample allowed us to infer the properties of the data set accurately, so most of the evaluation process was performed with the sample instead.

To estimate precision, three reviewers decided independently whether each message was relevant or not, similarly to a binary classification problem. Table 2 shows reviewer interagreement and the computed precision for each. The values of interagreement and Fleiss's kappa indicate that the tagging task is well defined and that the annotations obtained are reliable enough to use the harmonic mean of the precision of each reviewer for further uses in the evaluation process.

Evaluation

Through the annotated sample, we estimate precision (ratio of relevant retrieved tweets) with an error less than 1%, so this is properly computed and included in this study. However, the computation of recall is a large issue in our case. To explain why the computation of recall is not feasible, we have to go back to its meaning: It measures how many relevant documents were retrieved against the total relevant documents that exist. In our case, this means we should know how many tweets exist related to the selected topic, which is not possible.

Furthermore, sampling is not a feasible method for obtaining an accurate estimation of the number of existing tweets related to the topic: Twitter provides a sample stream with a random distribution of messages, but it is extremely rare that a tweet related to the topic appears in the sample, because of the insignificant proportion of them compared with the total volume of Twitter messages.

In considering the highest key word set size in our experimental setup ($k = 20$), the resulting data set has roughly 3 million messages, and Twitter generates on average a billion messages every week, leaving little margin for obtaining a significant sample of relevant messages. Thus, we had to leave the proper recall measure aside in this evaluation process and use another measure.

Instead, we used what we call "data set recall," which is similar to the recall measure but is in the scope of the data set, taking into account only all the relevant retrieved tweets as an estimate of all relevant tweets. With this method, we can measure the relevant retrieved tweets with different parameters. Similarly to the precision measure, this measure can produce any value, $\in \mathbb{R}$: $x \in [0, 1]$.

Table 3 shows a comparison of the performance results of the method against the baseline and the typical ad hoc static term-set method used in most of the applications. The proposed terms for the static method were ones having a high relevance within the Spanish community referring to the event (#vamosespaña, #eurocopa, #eurocopa2012, #laroja), whereas, for the our dynamic method, we showed values of $k = 10$ and $k = 20$.

Overall, it is observed that our method yields a higher data volume with a low precision loss, but the selection of k is very nuanced and very problem dependent. The precision loss is due in part to the unwanted inclusion of users with a huge number of followers, whose messages have a high impact in the social network but are highly noisy because they write messages about many different topics and are mentioned by many users.[1]

[1]When querying Twitter with a user name, we are retrieving not only tweets from that user but also those tweets that mention him.

Analysis of Results

One way to analyze the performance of the method effectively is to observe its behavior regarding the size of the dynamically computed keyword set, reflecting how good the queries being generated are and what effects they have on the resulting data set.

Keyword Set Size

Figure 5 shows the precision with different term-set sizes (k), and it is easy to observe the precision loss associated with the increase in the term-set size. This behavior is not unexpected because the method tries to increase the volume of the data set by introducing new terms, and this process usually introduces some noise. Nevertheless, this drop in precision is relatively low, with great stability in some intervals of the keyword-set size parameter.

As mentioned previously, the computation of the recall measure was not feasible, so we measured the data set recall instead. This is not a real substitute for the recall measure, but we think that the data set recall measure is quite appropriate, because the main focus of the method is increasing the volume of the data gathered without incurring a high precision loss. Figure 6 shows the behavior of the data set recall with different term-set sizes.

A better way to identify the best k is arranging both metrics similar to a Pareto frontier figure; Figure 7 gives an example of this arrangement. It shows an "efficiency" curve, each point being the combined precision and volume for each k value, and it is observed that for $k = 8$ the curve exhibits a good trade-off between precision and volume.

That point achieves a precision of 93.43% and a data set recall of 74.78%, resulting in an increase of 5.32 times the original volume while lowering the precision by only about 3.5%. Furthermore, the curve starts to decrease greatly and might inconveniently increase the term set size. Of course, this is not a general conclusion; the best value of k may depend on the nature of the topic selected.

Pruning Noisy Users

As mentioned previously, some users greatly contributed to the observed precision loss, introducing a great quantity of noise. These users are usually famous people, two good examples in our data set being the Spanish Formula One racer Fernando Alonso (@alo_oficial) and the Spanish singer-songwriter Alejando Sanz (@alejandrosanz). These users were initially selected by the method because they cheered up the Spanish national football team, but they were retained because of their high importance. About 44.62% of the collected messages related to @alejandrosanz are related to the topic explored, generating a considerable amount of noise, but, in the case of the user @alo_oficial, only 15.80% of tweets are related to the topic, most of the messages being a source of high noise.

If we remove these noisy users from the data set in a hypothetical filtering step, we see an improvement in the precision with some volume loss because this precision rise is quite significant. If we filter out these noisy users, our

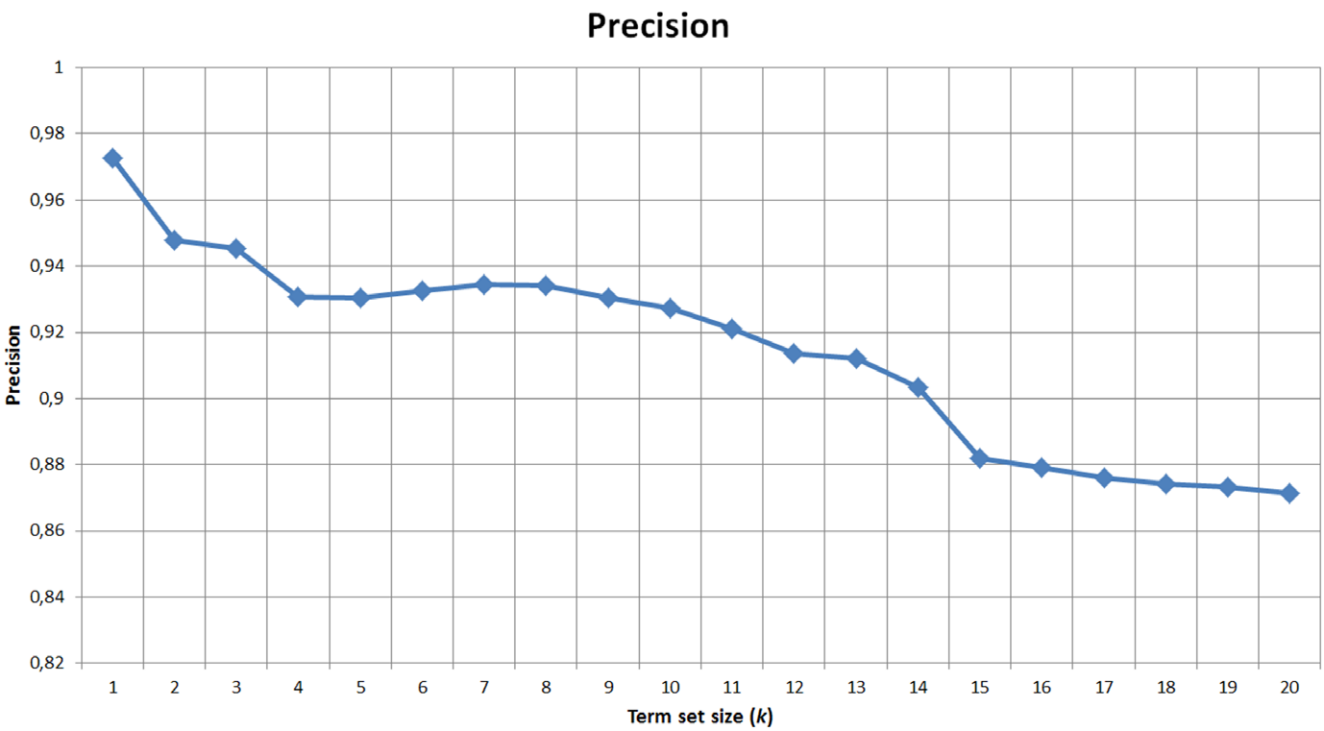

FIG. 5. Precision of the data set regarding the term set size (k). [Color figure can be viewed in the online issue,which is availableat wileyonlinelibrary.com.]

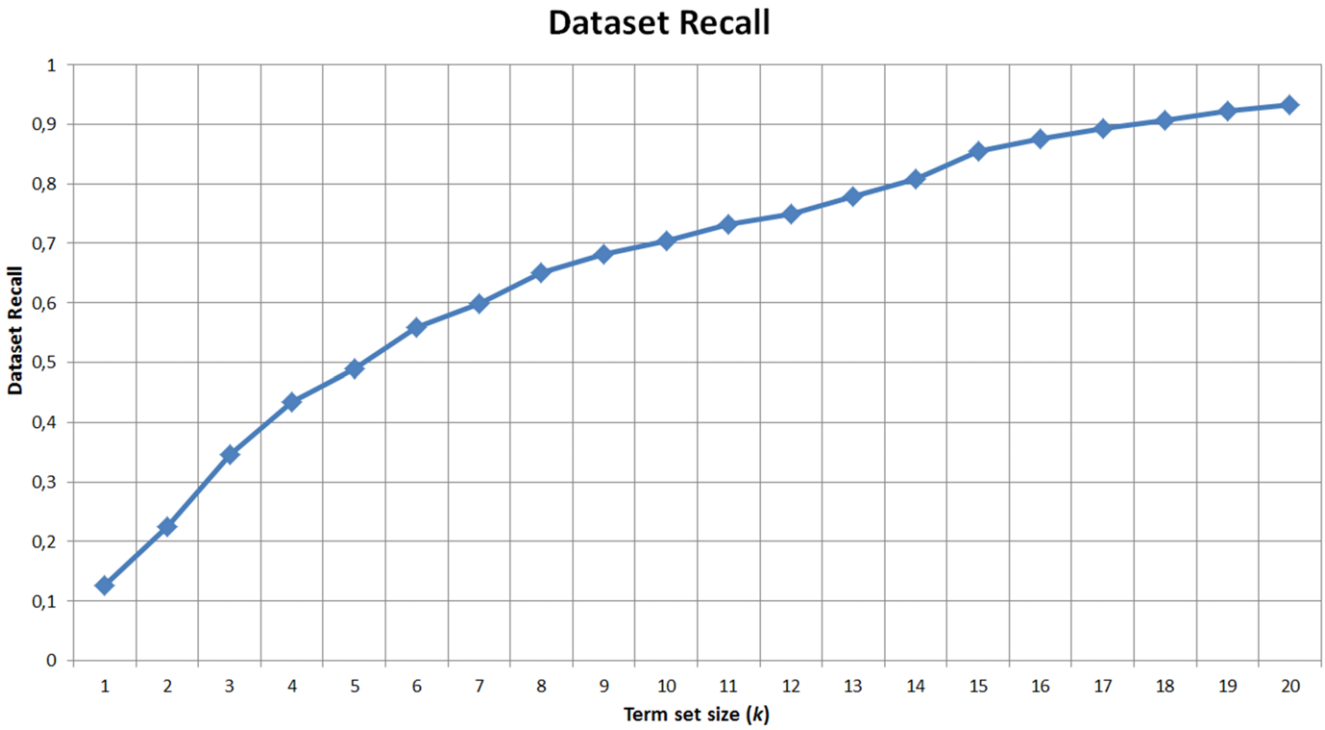

FIG. 6. Data set recall regarding the term set size (k). [Color figure can be viewed in the online issue,which is availableat wileyonlinelibrary.com.]

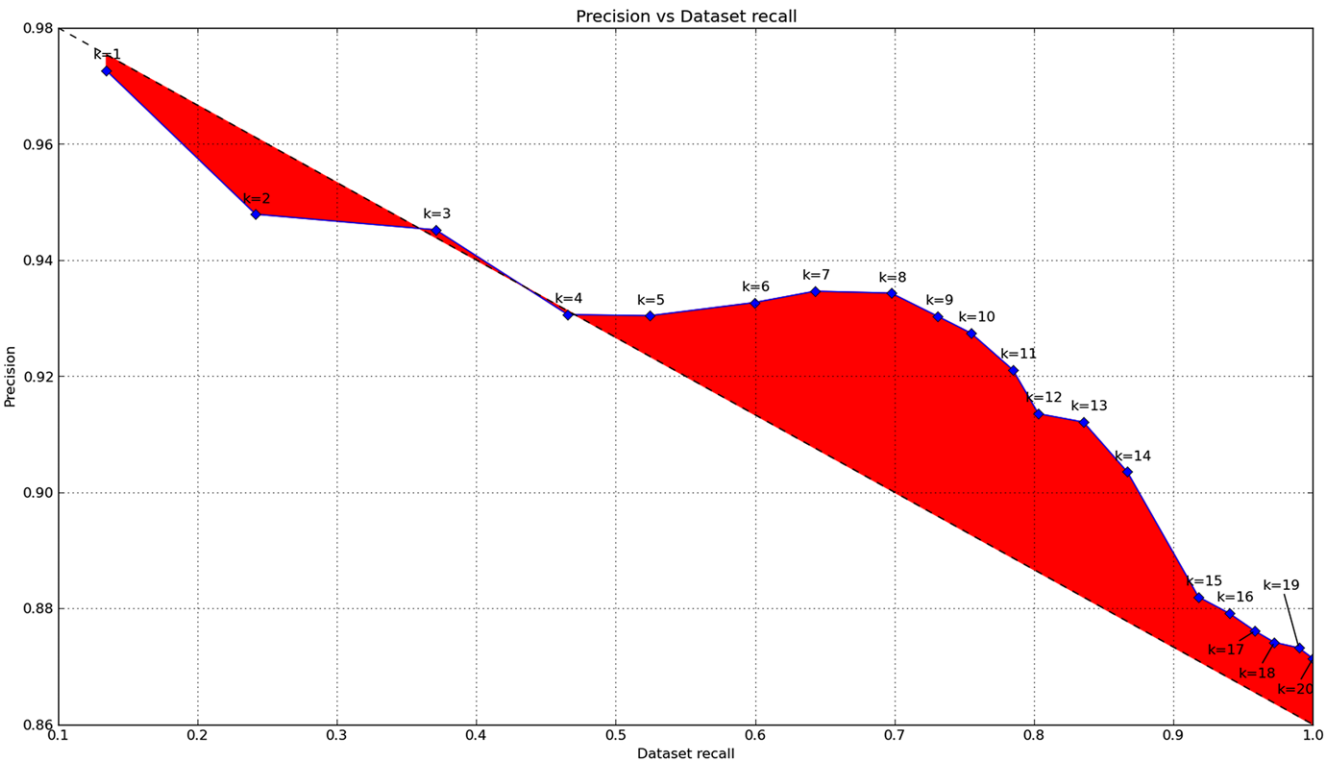

FIG. 7. Combined precision and data set recall for each k, showing a Pareto frontier curve. [Color figure can be viewed in the online issue,which is availableat wileyonlinelibrary.com.]

method raises the precision to higher than 91% when using $k = 20$ and to higher than 96% using the proposed $k = 8$.

Table 4 gives a method comparison including the mentioned filtering step and shows the data set recall and precision difference between filtered and nonfiltered versions of the method. Figure 8 shows the relevance graph corresponding to the match between Spain and Portugal during the Euro2012 event; it is visually easy to detect the most relevant entities in the graph and their interconnectivity. Because the central term (#vamosespaña) is relevant and highly interconnected with most of the graph, it is omitted for clarity's sake.

TABLE 4. Precision and data set recall comparison between methods including noisy users filtering.

Method	Precision	D. Recall	Prec. diff	D. Recall diff
$k = 8$	0.9343	0.7478	—	—
$k = 10$	0.9274	0.8034	—	—
$k = 20$	0.8714	1.0000	—	—
$k = k = 8$ w/filtering	0.9604	0.8298	+2.61%	−1.74%
k = 10 w/filtering	0.9521	0.8609	+2.48%	−1.77%
$k = 20$ w/filtering	0.9185	0.9488	+4.72%	−1.88%

The user @alejandrosanz exhibits the behavior of an "exterior hub," having a large community but being loosely connected with the rest of the terms, indicating the small relationship to the rest of the graph. This correlates highly with the noise previously related to this user, and it is straightforward that cutting off this kind of exterior hub from the graph may be a good approach in the filtering step.

Conclusions and Future Work

We provided a formalization for a previously undefined task that we believe is needed now: the topic-related tweet retrieval task. From this definition, we devised and proposed a general method addressing the task, focusing on analyzing the structural information of the underlying graph. Also, we showed the effectiveness of the method and highlighted some considerations regarding parameter selection and the noise elements in the problem.

As regards future work, we are considering more types of relationships for their use in the graph-building phase and devising some modifications to the PageRank algorithm that

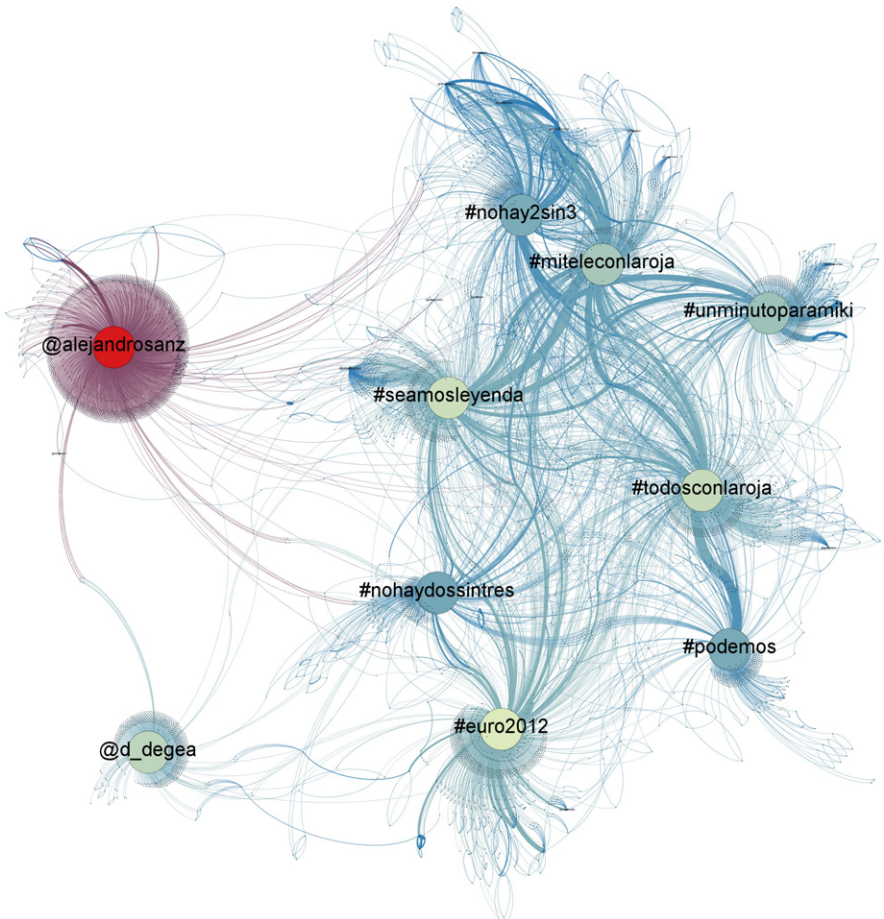

FIG. 8. Relevance graph corresponding to the match between Spain and Portugal during the Euro2012 event. [Color figure can be viewed in the online issue,which is availableat wileyonlinelibrary.com.]

may be useful in the relevance computation. Also, we are working on addressing the filtering subtask, performing content analysis as a complementary approach to structural analysis, but analyzing the text of a tweet is another problem in itself.

References

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of Twitter data. In Proceedings of the Workshop on Language in Social Media (LSM 2011) (pp. 30–38). Portland, OR: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology-new/W/W11/W11-0705.bib.

Cao, G., Nie, J.-Y., Gao, J., & Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 243–250). New York: ACM. Retrieved from http://doi.acm.org/10.1145/1390334.1390377 doi: 10.1145/1390334.1390377

Congosto, M.L., Fernández, M., & Egido, E.M. (2011). Twitter y política: Información, opinión y predicción? Cuadernos de Comunicación Evoca, 4.

Davidov, D., Tsur, O., & Rappoport, A. (2010a). Enhanced sentiment learning using Twitter hashtags and smileys. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters (pp. 241–249). Stroudsburg, PA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1944566.1944594.

Davidov, D., Tsur, O., & Rappoport, A. (2010b). Semi-supervised recognition of sarcastic sentences in Twitter and Amazon. In Proceedings of the 14th Conference on Computational Natural Language Learning (pp. 107–116). Stroudsburg, PA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1870568.1870582

Gayo-Avello, D., Metaxas, P.T., & Mustafaraj, E. (2011). Limits of electoral predictions using Twitter. In Proceedings of the fifth international AAAI conference on weblogs and social media. Retrieved from http://cs.wellesley.edu/%7Epmetaxas/ICWSM11-limits_predict_elections.pdf.

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. In Processing. Retrieved from http://www.stanford.edu/~alecmgo/. . ./TwitterDistantSupervision09.pdf

Golbeck, J., & Hansen, D. (2011). Computing political preference among Twitter followers. In Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems (pp. 1105–1108). New York, NY: ACM. Retrieved from http://doi.acm.org/10.1145/1978942.1979106 doi: 10.1145/1978942.1979106.

Hong, S., & Nadler, D. (2011). Does the early bird move the polls? The use of the social media tool "Twitter" by U.S. politicians and its impact on public opinion. In Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times (pp. 182–186). New York, NY: ACM. Retrieved from http://doi.acm.org/10.1145/2037556.2037583 doi: 10.1145/2037556.2037583.

Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011). Target-dependent Twitter sentiment classification. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Vol. 1 (pp. 151–160). Stroudsburg, PA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=2002472.2002492

Kim, E., Gilbert, S., Edwards, M.J., & Graeff, E. (2009). Detecting sadness in 140 characters: Sentiment analysis of mourning Michael Jackson on Twitter. Web Ecology, 3. Retrieved from http://www.mendeley.com/research/detecting-sadness-in-140-characters-sentiment-analysis-of-mourning-michael-jackson-on-twitter/.

Lavrenko, V., & Croft, W.B. (2001). Relevance based language models. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information retrieval (pp. 120–127). New York, NY: ACM. Retrieved from http://doi.acm.org/10.1145/383952.383972 doi: 10.1145/383952.383972.

Lee, K.S., Croft, W.B., & Allan, J. (2008). A cluster-based resampling method for pseudo-relevance feedback. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 235–242). New York, NY: ACM. Retrieved from http://doi.acm.org/10.1145/1390334.1390376 doi: 10.1145/1390334.1390376

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report No. 1999-66, Stanford InfoLab. Retrieved from http://ilpubs.stanford.edu:8090/422/ (previous number SIDL-WP-1999-0120).

Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). Valletta, Malta: European Language Resources Association (ELRA).

Pennacchiotti, M., & Popescu, A.M. (2011). Democrats, Republicans and Starbucks afficionados: User classification in Twitter. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 430–438). New York, NY: ACM. Retrieved from http://dx.doi.org/10.1145/2020408.2020477; doi: 10.1145/2020408.2020477

Silva, I.S., Gomide, J., Veloso, A., Meira, W., Jr., & Ferreira, R. (2011). Effective sentiment stream analysis with self-augmenting training and demand-driven projection. In Proceedings of the 34th International ACM SISGIR Conference on Research and Development in Information (pp. 475–484). New York, NY: ACM. Retrieved from http://doi.acm.org/10.1145/2009916.2009981 doi: 10.1145/2009916.2009981

Soboroff, I., Ounis, I., & Lin, J. (2012). Overview of the trec-2012 microblog track. In The 21st Text Retrieval Conference Proceedings.

Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., & Li, P. (2011). User-level sentiment analysis incorporating social networks. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1397–1405). New York, NY: ACM. Retrieved from http://doi.acm.org/10.1145/2020408.2020614 doi: 10.1145/2020408.2020614

Tao, T., & Zhai, C. (2006). Regularized estimation of mixture models for robust pseudo-relevance feedback. In Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval (pp. 162–169). New York, NY: ACM. Retrieved from http://doi.acm.org/10.1145/1148170.1148201 doi: 10.1145/1148170.1148201.

Tumasjan, A., Sprenger, T.O., Sandner, P.G., & Welpe, I.M. (2010). Predicting elections with Twitter: What 140 characters reveal about political sentiment. Word Journal of the International Linguistic Association, 178–185. Retrieved from http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/viewFile/1441/1852