A Summarization Tool for Time-Sensitive Social Media

Walid Magdy, Ahmed Ali, and Kareem Darwish
Qatar Computing Research Institute
Qatar Foundation
Doha, Qatar
{wmagdy, amali, kdarwish}@qf.org.qa

ABSTRACT

Searching social content in general and microblogs (aka tweets) in particular has been basic and limited, especially for time-sensitive topics. The currently implemented microblog search on sites such as Twitter is based on simple word matching and retrieves the most recent microblogs that match a given query. Furthermore, a user may obtain hundreds or perhaps thousands of microblogs in response to a given query, leading to information overload. We present a new multidimensional microblog search tool that generates a comprehensive report from microblogs instead of a flat list of recent/relevant microblogs for a given query. Reports may include tag-clouds, topic time series, and most popular and funny microblogs, etc. The tool can be configured for monitoring time-sensitive topics using a set of predefined queries. We demonstrate our system on Arabic and English microblog collections. Additionally, we show a special configuration of the system for monitoring the 2012 Egyptian presidential elections.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing, H.3.3 Information Search and Retrieval

General Terms

Algorithms, Design, Human Factors, Languages.

1. INTRODUCTION

Many microblog and social websites, such as Twitter, provide search capabilities to allow users to find relevant posts that match their information need. The currently implemented microblog search on Twitter provides recent tweets that match search words. A user may elect to search (or follow) for specific entities, persons, or events, via the use of hashtags "#tag" or name mention "@user", to get continuous updates [10]. One disadvantage of this kind of search is that a query may yield a large number of tweets, overwhelming the user. In this scenario, a user is presented with a flat list of matching tweets (tweets and microblogs are used interchangeably in the paper), leaving much to be desired.

In this demonstration, we present a microblog search tool that generates a comprehensive report in response to a given query. A user searches for a particular time-sensitive topic such as election, a sports event, or natural phenomenon, or an entity such as a person, location, organisation, or product. The user chooses a time span to get a summarized but comprehensive report from the public tweets about the topic from Twitter. For the resulting tweets in the specified time-span, the report produces the top tweets (top in the paper means most (re)tweeted), top funny tweets, most circulated videos and links, most popular terms and phrases, and statistics about the entity/event over time. A user can also navigate through the resulting report over time to see how the popularity of a given entity/event has changed. In addition, our system can be configured for automatically collecting tweets related to a given topic to monitor special events for a period of time. Section 4 presents a configuration of the system to monitor the 2012 Egyptian presidential elections and to present daily reports for the elections in general and for each of the candidates in specific.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

2. MICROBLOG RETRIEVAL

Interest in microblog retrieval has significantly increased in recent years. Several studies investigated the nature of microblog search compared to other search tasks [6, 10]. Naveed et al. [6] illustrated the challenges of microblog retrieval, where documents are very short and typically focus on a single topic. Taveen et al. [10] highlighted the differences between web queries and microblog queries, where microblog queries usually represent users' interest to find updates about a given event or person as opposed to finding relevant pages on a given topic in web search.

Due to this increased interest in microblog search, TREC introduced a new track focused on microblog retrieval in 2011 [7]. The aim was to find the best methods for achieving high precision retrieval for microblog search. A collection of 14 million tweets from Twitter and a test set of 50 topics were provided for investigation [7]. Although the track led to a variety of effective retrieval approaches, the issue of modeling the search scenario remains important as the TREC track setup models search like a standard ad-hoc retrieval task, which may be suboptimal [10].

The absence of a sensible definition for a microblog search scenario led some researchers to create different useful tasks other than direct search. For example, [9, 11] used tweets as a news source and compared them to other online news media to detect features for automatic news detection from Twitter. In [8], tweets were used to recommend news to users based on their preferences. In [1], users' mood on Twitter was utilized to predict stock market changes. Many other tasks have been suggested for achieving information gain to users based on social data from Twitter.

In this paper, we present a user experience that is different from the tweet search that is available from Twitter.

3. SYSTEM ARCHITECTURE

3.1 Basic System Architecture

Figure 1 presents the basic system architecture of our system. Tweets from Twitter are collected for a given language and saved in a database. Tweet text is then normalized using advanced text normalization techniques for slang language that is used in tweets, for English as in [3] and for Arabic as in [10]. Normalized tweets are then indexed along with their metadata, such as author ID,

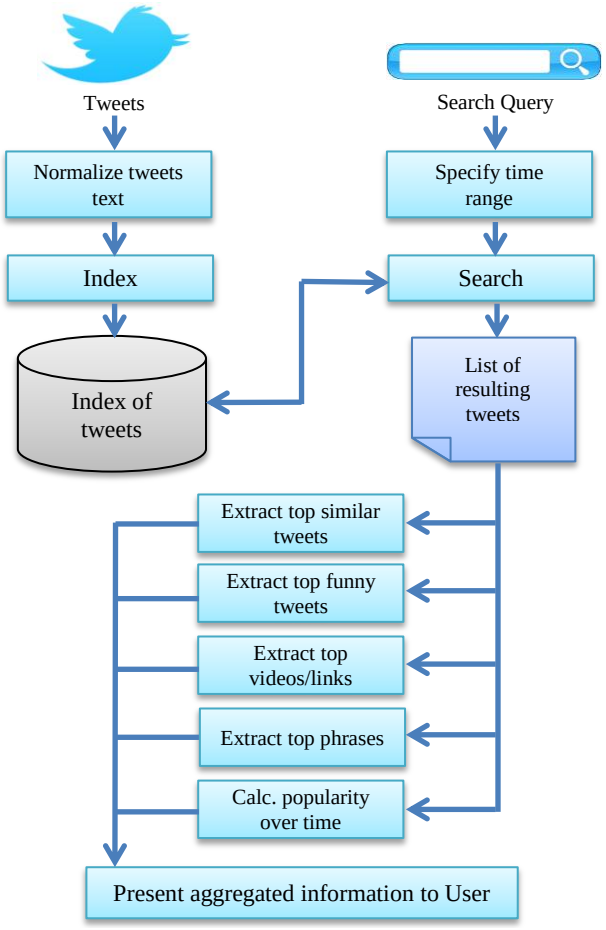

Figure 1 Block diagram of the system

time stamp, and tweet ID. A user provides a search query, which would preferably be an entity or an event, or could be a hashtag (#tag), a name mention (@some_user), or a free-form query. The user could limit search results to a specific time window. Otherwise, the default time window is the current day starting from 12:00 am. All resulting tweets in the specified time window are retrieved, and the following information is extracted from the retrieved tweets:

1. Top tweeted messages, where we allow for limited textual variation in the phrasing of tweets.
2. Top tweeted messages that contain funny emoticons.
3. Most circulated videos and links in the tweets.
4. Most frequent terms/phrases appearing in tweets.
5. The popularity of the search topic over time.

All extracted information from the retrieved tweets is then presented in a user-friendly summarized form, where top tweets, top funny tweets, and most circulated videos and links are sorted by frequency of appearance. Most frequent terms and phrases are presented in the form of tag-cloud. A time-series graph shows the popularity of the topic on Twitter over time.

We believe this form of presentation to search results can lead to higher information gain compared to standard list of results that are usually very large and users cannot get clear information from.

3.2 System Components

3.2.1 Collecting Tweets

Tweets are collected by issuing generic queries, such as "lang:xx" (ex. "lang:ar" for Arabic) against Twitter, which retrieves tweets in a given language. We used *tweet4j* package for collection. Collected tweets contain the author and tweet ID's, timestamps, etc. We stored all collected tweets in a database.

3.2.2 Normalizing Tweets Text

We preprocessed the collected tweets using language dependent normalization modules. Tweet text is characterized by its special style, where the language used contains slang and dialectic forms of words. We used dialectic Arabic normalization for Arabic tweets as in [5] and used a simpler implementation of the English normalization technique presented in [3] for English tweets. We normalized emoticons that refer to similar happiness or sadness emotion as in Table 1.

Table 1 Normalizing happy/sad emotion words

:)	:), :-), :-)), :D, :d, ^_^, lol, loool, hahaha, …
:(	:(, :-(, :--((, ;(, :'(, …

3.2.3 Indexing and Searching Tweets

We indexed normalized tweets along with their metadata using Apache SOLR (ver. 4.0)[1], which is built on top of Lucene. We configured SOLR to use Boolean retrieval model [4]. We used a Boolean model instead of a ranking model since we are interested in "all" tweets that match a query in a given time window.

3.2.4 Extracting Top Tweets and Top Funny Tweets

All retrieved tweets for a given search query are then grouped to aggregate all similar tweets into the same group. For a fast and robust matching between tweets, we applied an additional normalization step, which involves case-folding and removal of all hashtags, name mentions, URLs, punctuations, symbols, emoticons, and retweet symbols. Tweets that match exactly after normalization are grouped together. Groups were presented in ranked order (in descending order) by their size with the most common tweet form as the representative of the group along with the number of tweets in the cluster. Top funny tweets were extracted in the same manner, but the clustering was applied to those tweets that have smiley emoticons only.

3.2.5 Extracting Top Terms/Phrases

For Arabic, we used a base-phrase chunker that is akin to AMIRA [2] to extract noun-phrase. For English, we used Open Calais to extract keywords/keyphrase. We sorted extracted noun-phrases and/or keywords/keyphrases by their frequency, and we displayed them in a tag-cloud. We allowed the inclusion of hashtags and name mentions, but we excluded URLs.

3.2.6 Extracting Top Videos/Links

The URLs in the tweets of the top 100 clusters are extracted. Since URLs in tweets are typically shortened and some URLs may have multiple shortened forms, we expanded all URLS to find the original URLs. We used URLs pointing to YouTube videos to obtain a ranked list of the most popular videos so we can embed them in the resultant report. Other URLs are extracted and their titles are presented ordered by the number of appearances in tweets along with their links and number of appearance.

3.2.7 Navigation over Time

The number of tweets across time is plotted and presented to the user in an interactive graph. The time unit used in our system was a day, but the system can be configured for other time units. Also, the user has the option to explore the resulting report day-by-day and navigate through the specified period of time to see a summary for each day individually.

[1] http://lucene.apache.org/solr/

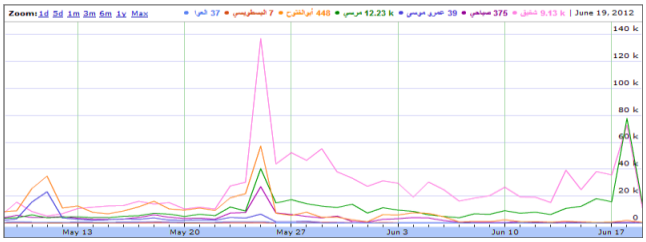

Figure 2 Popularity of candidates over time

3.3 Pre-Configuring for Special Events

The system described in this paper can be used for tasks beyond searching for a given topic on Twitter. It can be configured to monitor the popularity of specific entities or events over time. The system is fed with a set of fixed queries, and summarized reports are updated continuously at fixed time intervals to provide users with updated reports.

Queries used for the system can be rich Boolean queries or can be based on text classification. Boolean queries, though require time to manually construct, do not require training and can help disambiguate entities or events than may be referred to in multiple topics. For example, searching for the French president "Hollande" can retrieve many tweets referring to different persons carrying the same name. The Boolean query can be formulated as: "*Hollande* AND (*François* OR *France* OR *president*)" to disambiguate the entity. Using text classification would require a training using a few dozen positive and negative examples for the entity or event, and a trained classifier, such as a support vector machine classifier, can effectively produce high precision results.

Multiple entities can be monitored within a given event, and the relation among these entities can be extracted and plotted in graphs to show the connection among different entities.

4. DEMONSTRATION

We present two illustrative demos of our system; the first one is a configuration of our system for automatically monitoring the tweets on the Egyptian presidential election from Arabic tweets. The second is a free search in English and Arabic tweets to demonstrate the effectiveness of the resulting report.

4.1 Monitoring the Egyptian Presidential Elections

Our system has been configured to monitor the 2012 Egyptian presidential election on Twitter. This election has grabbed much attention in the Arab world as the first such election after the Egyptian revolution in 2011. We created a website called (*www.rayesna.com*) – meaning "our president" in Arabic – for presenting summary reports on the Egyptian tweets about the elections. The website allows users to see daily reports on all the presidential candidates based exclusively on tweets, including all the features mentioned earlier. It also provides the relation between all candidates and plots of their popularity over time.

4.1.1 Data Collection

The website used a collection of Arabic tweets that we started collecting from February 26, 2012, which is 3 months before the elections day. Roughly, we were collecting 2.6 million Arabic tweets per day. The number of tweets used in daily reports varied dramatically over the three months period when we were monitoring the election. The number of tweets used in reports ranged between 6k tweets at the end of February up to 377k and 158k tweets on election days of the first and second election rounds respectively. Figure 2 shows the time series for all candidates' occurrences on Twitter, as presented on the website. The graph has two peaks on May 25, the first round election-day between 13 candidates; and on June 17, the second round election-day between the two frontrunners in round 1.

The daily reports are constantly updated using newly found tweets every 10 minutes and aggregation is done day-by-day, where we set days to start from 12 midnight.

4.1.2 Configuring the system

- *Creating rich queries*

A set of rich Boolean queries were prepared for collecting the tweets, including queries for 13 candidates running for president and some other entities related to the election, such as people who were expected to run the elections but did not, top political parties, and governmental organizations related to the elections.

Some of the prepared queries were just the name of the candidate or entity, while others required rich Boolean queries to disambiguate them. For example, the candidate "صباحي – (Sabahi)" required a complex query, because his name also means "my morning" in Arabic, yielding many unrelated tweets.

- *Calculating candidates popularity and relations*

The popularity of each candidate was measured by the count of tweets that includes his name. We noticed that the most popular candidate has always been the object of negative and sarcastic tweets, while the second most popular has usually been one of strong candidates with positive/supportive comments.

Relation between candidates was measured by the co-occurrence of candidates in tweets. Anecdotally, we observed that many weeks before election day, co-occurrences of candidate mentions in tweets were more frequent among candidates that are ideologically similar; then as election day drew closer, co-occurrences were based on sub-events such as political debates, mass media appearances, etc.

4.2 Free Search on Tweets Collections

During this demo users will be able to search tweets over the Arabic collection and over a sample of recent English tweets. The user should specify a search query and a time window to get a summary report about what is mentioned about the given query on twitter in the specified window of time.

5. REFERENCES

1. J. Bollen, H. Mao, X-J. Zeng. (2010). Twitter mood predicts the stock market. *Journal of Computational Science. 2(1)*
2. M. Diab. (2009). Second generation tools (AMIRA 2.0): Fast and robust tokenization, POS tagging, and base phrase chunking. *MEDAR 2009.*
3. B. Han, T. Baldwin. (2011). Lexical Normalisation of Short Text Messages: Makn Sens a #twitter. *ACL-HLT 2011.*
4. F. W. Lancaster, E. G. Fayen. (1973). Information Retrieval On-Line. *Melville Publishing Co., Los Angeles, California*
5. W. Magdy, A. Ali, K. Darwish. (2012). Language Processing for Arabic Microblog Retrieval. *CIKM 2012.*
6. N. Naveed, T. Gottron, J. Kunegis, A. Alhadi. (2011). Searching microblogs: coping with sparsity and document quality. *CIKM-2011.*
7. I. Ounis, C. Macdonald, J. Lin, I. Soboroff. (2011). Overview of the TREC-2011 Microblog Track. *TREC-2011.*
8. O. Phelan, K. McCarthy, M. Bennett, and B. Smyth. (2011). Terms of a feather: content-based news recommendation and discovery using twitter. *ECIR 2011.*
9. I. Subasic, B. Berendt. (2011). Peddling or Creating? Investigating the Role of Twitter in News Reporting. *ECIR-2011*
10. J. Teevan, D. Ramage, M. Morris. (2011). #Twittersearch: A comparison of microblog search and web search. *WSDM 2011.*
11. W. X. Zhao, J. Jiang, Ji. Weng, J. He, E-P. Lim, Ho. Yan, X. Li. (2011). Comparing twitter and traditional media using topic models. *ECIR 2011.*