QweetFinder: Real-Time Finding and Filtering of Question Tweets

Ameer Albahem[1(✉)], Maram Hasanain[2], Marwan Torki[2], and Tamer Elsayed[2]

[1] RMIT University, Melbourne, Australia
ameer.albahem@rmit.edu.au
[2] Department of Computer Science and Engineering, Qatar University, Doha, Qatar
{maram.hasanain,mtorki,telsayed}@qu.edu.qa

Abstract. Users continuously ask questions and seek answers in social media platforms such as Twitter. In this demo, we present QweetFinder, a Web-based search engine that facilitates finding question tweets (Qweets) in Twitter. QweetFinder listens to Twitter live stream and continuously identifies and indexes tweets that are answer-seeking. QweetFinder also allows users to save queries of long-term interest and pushes real-time qweet matches of saved queries to them via e-mail.

1 Introduction

Users turn to social media platforms, such as Twitter, to explore developments of topics and events and to seek opinions of others on matters of interest. Consider a user that is following-up on a running topic like "US election debate". She might be interested in answers of several on-topic questions including subjective ones, e.g., "Who will win the debate?", or more factual ones, e.g., "When is the next debate?". In many cases, users believe that the Web might not have satisfactory answers to some "real-time" questions, however, tweeters might be able to answer them more effectively [4]. Furthermore, the user might not even know what the right questions to ask are, due to lack of full knowledge about the topic. In this work, we propose **QweetFinder**, a system that helps a user find questions on Twitter on a topic of interest. The task of suggesting or retrieving questions given a query is not new, yet the majority of existing systems were designed for community question answering (CQA) platforms [5]; up to our knowledge, only few systems were developed for query-oriented conversation retrieval directly from Twitter (e.g., [2]). Differently from those studies, our system is multilingual as it is currently designed to retrieve Arabic and English questions. In addition, we present an architecture to build such systems using mature open-source technologies.

A large body of literature has showed that users of social media platforms are regularly seeking information from others by posting questions [7]. Several studies focused on understanding the nature and types of such questions or on developing systems to answer questions posted on Twitter. We present an alternative view on how users with questions can interact with Twitter. We

J.M. Jose et al. (Eds.): ECIR 2017, LNCS 10193, pp. 766–769, 2017.
DOI: 10.1007/978-3-319-56608-5_79

propose a ***real-time*** system that suggests questions to users that are relevant to their topics of interest. Studies on CQA suggest that, in many cases, users attempt to find existing questions and answers on topics of interest before posting a new question [5]; we believe Twitter users can benefit from such service as well. The proposed system also enables the user to explore questions of different perspectives she might not have thought of initially. Moreover, a recent study found that, in many cases, users are reluctant to post some kinds of questions on their own social network to avoid disturbing their followers [4]. Our system offers the user the opportunity to anonymously explore existing questions (and their replies) without explicitly asking them; we believe such flexibility will encourage more users to turn to Twitter for answers on questions that are best asked there.

Use cases of QweetFinder are not only limited to those of interest to normal users. For example, QweetFinder can potentially be used by organizations for market research and collection of unsolicited user feedback on a brand or product. In a leading study, Jansen et al. [3] found that out of 2.7K tweets about different brands, 11% were information-seeking tweets. A company can benefit from such user questions for quality management and product improvement.

2 System Description

QweetFinder is a real-time multilingual query-oriented question retrieval and filtering system[1]. It retrieves answer-seeking question tweets (qweets). QweetFinder consists of two layers: the back-end engine and the Web-based front-end (i.e., user interface).

2.1 System Back-End Architecture

As shown in Fig. 1, the back-end layer consists of the following components:

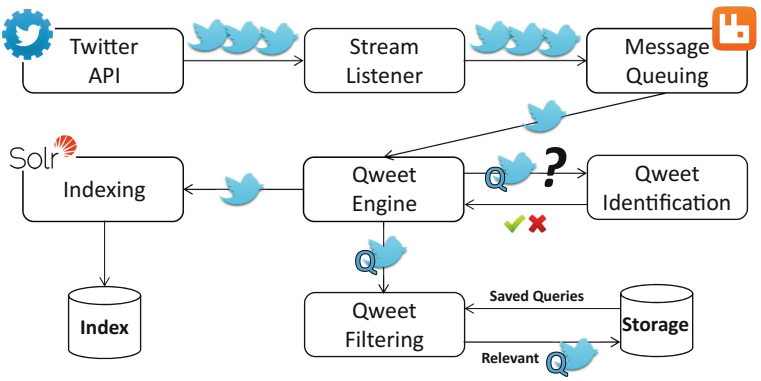

Fig. 1. QweetFinder back-end architecture

[1] A prototype of QweetFinder can be found at www.qweetfinder.com.

Tweet Streaming: Twitter provides a streaming API to directly listen to tweets as they are posted. We used Tweepy[2], a Twitter open source streaming library, to access the streaming API. In particular, we subscribed to all tweets that contain Arabic or English question marks, Arabic question phrases or English Wh-question words[3]. In addition, we limit the languages of tracked tweets to only Arabic and English.

Twitter requires streaming clients to process tweets as soon as they arrive and penalizes clients that fall behind. Since we need to perform qweet identification, which adds a time latency on processing the streamed tweets, we used a Message Queue component to tackle the expected latency and store the streamed tweets temporarily until they are consumed by the Tweet Processing Pipeline. We investigated different options such as Apache ActiveMQ and Amazon Kiness, Apache Spark and RabbitMQ. We found RabbitMQ[4] a good option in terms of scalability and learning curve.

Tweet Processing Pipeline: In this component, we developed a consumer that reads tweets from RabbitMQ and processes them through three steps:

1. **Qweet identification**: Not all questions in tweets are answer-seeking [1]. Therefore, we developed Arabic and English qweet classifiers to filter out qweets from a stream of tweets. Arabic qweet filtering was performed using SVM question classifier that leverages groups of features: lexical, structural, question-specific, tweet-specific, and (in)formality aspects of the tweets. The classifier was then trained on a manually annotated tweets collected through crowdsourcing—further details can be found in [1]. We classify English tweets using a Random Forest classifier using n-grams as features.
2. **Qweet Filtering:** Identified qweets are pushed to a filtering service (REST API) that filters them against a list of queries submitted by users. These qweets are then delivered to users via email. Qweet filtering can be done via different filtering mechanisms. In this demo, we used Luwak filtering engine[5]. Nevertheless, we plan to customize our participation at TREC-2016 Summarization Track [6] to suite the needs of qweet filtering.
3. **Indexing:** The last step is storing and indexing tweets. In our system, we used Apache Solr to index all tweets, flagging those that were classified as qweets by our qweet classifier.

2.2 User Interface

Users can interact with QweetFinder using a Web-based application implemented using the Play Framework[6]. The interface allows users to register with our system, search qweets, and save queries.

[2] http://www.tweepy.org/.
[3] Arabic phrases are available here: http://bit.ly/2itZDe9.
[4] Version 3.6.5.
[5] https://github.com/flaxsearch/luwak.
[6] https://www.playframework.com.

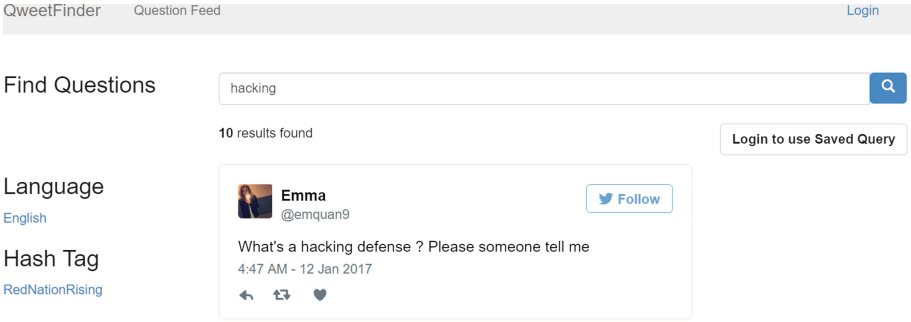

Fig. 2. A qweet search scenario

Figure 2 shows a user search scenario. An authenticated user issues a search query, then the system returns a list of qweets that match her query. Clicking a qweet will directly allow the user to view it in Twitter, as we assume that users might find answers to the qweet in the replies. The user can then save the query to his question feed. In the back-end, the new saved query is pushed to the filtering service to match incoming qweets against it. The system then pushes the new matched qweets to users via email.

Acknowledgments. This work was made possible by NPRP grant# NPRP 6-1377-1-257 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

1. Hasanain, M., Elsayed, T., Magdy, W.: Identification of answer-seeking questions in Arabic microblogs. In: Proceedings of CIKM 2014 (2014)
2. Herrera, J.M., Parra, D., Poblete, B.: Retrieving relevant conversations for Q&A on Twitter. In: Proceedings of SPS 2015, Co-located at SIGIR 2015 (2015)
3. Jansen, B.J., Zhang, M., Sobel, K., Chowdury, A.: Twitter power: tweets as electronic word of mouth. JASIST **60**(11), 2169–2188 (2009)
4. Oeldorf-Hirsch, A., Hecht, B., Morris, M.R., Teevan, J., Gergle, D.: To search or to ask: the routing of information needs between traditional search engines and social networks. In: Proceedings of CSCW 2014 (2014)
5. Srba, I., Bielikova, M.: A comprehensive survey and classification of approaches for community question answering. ACM Trans. Web **10**(3), 18:1–18:63 (2016)
6. Suwaileh, R., Hasanain, M., Elsayed, T.: Light-weight, conservative, yet effective: scalable real-time tweet summarization. In: TREC 2016 (2016)
7. Zhao, Z., Mei, Q.: Questions about questions: an empirical analysis of information needs on Twitter. In: Proceedings of WWW 2013 (2013)