BEST of KAUST at TREC-2011: Building Effective Search in Twitter

Jinling Jiang,[†] Lailatul Hidayah,[†] Tamer Elsayed,[‡] Hany Ramadan[†]
[†]Computer Science Department, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia
[‡]Microsoft Research Advanced Technology Lab, Cairo, Egypt
{jinling.jiang, lailatul.hidayah}@kaust.edu.sa, telsayed@microsoft.com, hany.ramadan@kaust.edu.sa

ABSTRACT

In our first-ever appearance at TREC, we explore initial ideas on building an effective search tool over tweet stream as a participation in this year's microblog track. Among those ideas are tweet expansion with short representation of terms that frequently co-occur with hashtags and URLs, and re-ranking based on statistics that estimate user popularity (using replies and mentions), tweet popularity, URL popularity and user topic authority (using simple user profiles). Initial results show that re-ranking improves the effectiveness while expansion sometimes harms it. Overall, the system built for the task is indeed a great resource for further extensions and experiments.

1. INTRODUCTION

[1] Twitter, an online microblogging service, is acting the leading role in this emerging form of new media. Within the 140 characters limit, users broadcast what they experience and think about their daily life and work activities by posting short snippets of text, called tweets [4]. According to the Twitter blog, fans sent 4,064 tweets in a single second when this year's Super Bowl came to its final moments[2], making the highest tweets per second (TPS) ever recorded for a sporting event. Based on a later report, Twitter users now send more than 140 million tweets a day[3]; this shows the huge popularity Twitter is gaining. People use Twitter for various reasons, such as serving as information source, keeping in touch with friends, and seeking information [1].

Searching Twitter is different from searching the Web. As investigated in [3], many people are interested in searching more timely and social information on Twitter than on the Web. Many events are lively broadcasted on Twitter and searching them is highly desired. People also search for social information related to other users, e.g., they might want to know who responded to some certain tweet or to find people with similar interests.

Features of Twitter also make the search different. The most significant feature of Twitter is the 140-character limit, while general web pages can be much longer. There are also underlying consequences, for example, tweets have only text while web pages can have multimedia, and tweets are more concentrated on their topics. Another feature is that the tweets are posted by authors, called twitterers, who might be related in the social graph by the "following" relationships. There is no such explicit social information on the web. Moreover, tweets are static while web pages are not: tweets do not change after being posted, but web pages can. These features should affect the design of the search system.

In this paper, we introduce BEST, which stands for "Building Effective Search in Twitter". The goal is to build a real-time ad-hoc search system for the official tweet corpus provided by TREC (Text REtrieval Conference) 2011 Microblog Track. The real-time aspect of the search concerns the requirement that the returned tweets should be both relevant to the query and recent to the issuing time. For the required run, no external information outside the official tweet corpus may be used. In all of our four submitted runs we adhered to that condition. Our system is implemented using Hadoop, the open source implementation of MapReduce distributed programming framework [11].

The rest of this paper is organized as follows. We first discuss the related work in Section 2. We then present our system design in Section 3. In Section 4, we conduct a series of evaluations of our system and finally conclude in Section 5.

2. RELATED WORK

In this section, we discuss the related work from the following aspects. We first discuss some investigations on Twitter that describe the user behavior and features of Twitter, followed by some current solutions of search in Twitter. We finally turn to the ranking of the search results, which is an important component in most search systems.

2.1 Twitter

Twitter has been more and more investigated on as it gains popularity over the Web. Researchers have asked "What" in [4, 5], "Why" in [2, 5], and "How" in [2, 3] regarding Twitter. The studies on "What" give a general picture of Twitter, like the features and constraints, user popularity, order of magnitude of number of tweets, etc. The studies on "Why" analyze the reason people use Twitter, classify the reasons into categories and also show the importance of Twitter. The studies on "How" reveal the way people use Twitter, which motivated in part some of the design decisions behind our system.

2.2 Searching Twitter

Currently, there have been a number of websites offering

[1] This work was performed while the third author was at King Abdullah University of Science and Technology (KAUST) in Thuwal, KSA.

[2] http://blog.twitter.com/2011/02/superbowl.html

[3] http://blog.twitter.com/2011/03/happy-birthday-twitter.html

the service of real-time micro-blogging search. Twitter[4] itself has search service based on the number of re-tweets while Tweetfind [5] ranks search results according to authority of authors. Bing[6] has also published twitter search service considering account authority and freshness of the tweets. Users can always utilize this kind of service to get tweets related to interesting topic via user-defined query. A recent paper [6] proposed a tweet index called TI, which indexes the tweets that are of high chance to appear as a search result while delaying the indexing of some other tweets. As a trade-off between the cost of online indexing and the quality of the search results, this paper manages not to compromise the latter.

2.3 Ranking

As an important component in search systems, ranking functions have been discussed extensively in the context of tweet search design. Here we select some of the ranking functions to demonstrate the idea. Chen at el. [6] proposed a new ranking scheme by combining the relationship between the user and the tweets. The tweets are grouped into topics and the ranking of the topics are updated dynamically. TwitterRank, proposed in [7], suggested a topic-sensitive influence measure to twitterers. The measure is an extension of PageRank and takes both the topical similarity between users and the link structure into account. Duan et al. [8] proposed a ranking method that involves multiple features, including the content relevance of a tweet, twitterer authority, embedded URL, etc. and learns the best set of features to use in the rank. Two challenges that are not encountered in traditional web search are introduced here: quickly crawling relevant content and ranking documents with impoverished link and click information. Our system also puts emphasis on the retrieval of very fresh (up-to-date) content. Dong et al. [9] advocated a method to use the micro-blogging data stream to detect fresh URLs and employ micro-blogging data to compute novel and effective features for ranking fresh URLs.

3. SYSTEM OVERVIEW

We identify four forms of connections in Twitter. First, posting a tweet naturally connects it to its twitterer. Second, re-tweets (i.e., resending/reposting an existing tweet which is remarked by ”RT”), replies (to an already posted tweet) and mentions (i.e. directing a tweet to specific twitterer(s) by using the @ convention) connect tweets with other tweets. Third, the URLs mentioned in tweets connect the tweets and the corresponding linked external web pages. Finally, the “follower graph” (explicitly constructed over the “following” relationships) connect twitterers.

As a typical information retrieval system, our system consists of preprocessing, indexing, and retrieval components.

3.1 Preprocessing

The preprocessing step is mainly concerned with filtering out undesired tweets and preparing the desired ones for indexing. We first remove null tweets and non-English ones (since the task is only concerned with English tweets for this year). We utilized the Java Text Categorizing Library (JTCL)[7] to implement language detection algorithm which is based on the technique described in Cavnar and Trenkle [10]. After this step, we get 6,035,601 English tweets. For detecting spam users, we borrowed the idea from [9] and defined the behaviors of spammers as the following: when one posted a URL at least n times or if one posted at least m URLs at least p times each. Experimentally, we set n,m,and p as 5,3,and 3 respectively and also removed the tweets of all of the detected spammers. There are 2538 spam users detected in our system this way. The survived tweets are then tokenized and stemmed. Each token is assigned a type (i.e., “retweet”, “mention”, “reply”, “hashtag”, or “other”). Later in the pipeline we need to calculate some statistics about URL popularity, tweet popularity, and user popularity, as explained in Section 3.3, so we built a database to store needed properties of tweets in this step.

3.2 Indexing

One of the challenges of the task is the tight length limitation of the tweets, which makes them very short and hence sometimes hard to understand in isolation. To tackle this problem, we proposed to expand the tweets in three different ways. The first one involves a tweet property called “hashtag”. The hashtag (which is denoted by a # followed by a tag) is a form of reflecting the topic (or the context) of the tweet (e.g., #jan25). We aim to expand a tweet that includes a hashtag by appending a short representation of the hashtag. To represent a hashtag in terms, we choose the terms that co-occur with it most frequently. Secondly, we do the same way of expansion with URLs that are mentioned in the tweets. We build two indexes that map hashtags and URLs to their frequently co-occurred terms. Both indexes are used to expand tweets that include hashtags and/or URLs by the corresponding expansion terms on the fly at indexing time. We also build a short profile for each twitterer that includes the most frequent terms appear in his/her tweets. We then build a third index that maps terms that appear in any of these profiles to their corresponding twitterers. Given query terms, we compute authority scores for twitterers using this index. These three indexes are build in addition to the conventional inverted index that associates terms with the tweets in which they occur.

3.3 Ranking and Retrieval

Given a query, we rank tweets in two steps. The first one (called primary ranking) reflects both the content similarity (using traditional retrieval models) and tweet recency (computed relative to the timestamp of the query). For the content similarity, we choose IDF instead of TF-IDF since term frequency may give out a misleading effect given the short length of the tweets. For the tweet recency, we only pick up the 10000 most recent tweets before the query timestamp. At the end of this stage, we get 100 most similar tweets. The second step re-ranks the top tweets resulted from the first stage using a list of computed statistics that has been made possible by querying a database. The computed features are: user popularity, URL popularity, tweet popularity, and user authoritative score. User popularity was initially supposed to reflect the popularity of the user using the follower graph (either by computing PageRank-like score for each twitterer, or simply as the number of followers). Unfortunately, the “following” relationships are not included in

[4] http://search.twitter.com/

[5] http://www.tweetfind.com

[6] http://www.bing.com

[7] http://textcat.sourceforge.net

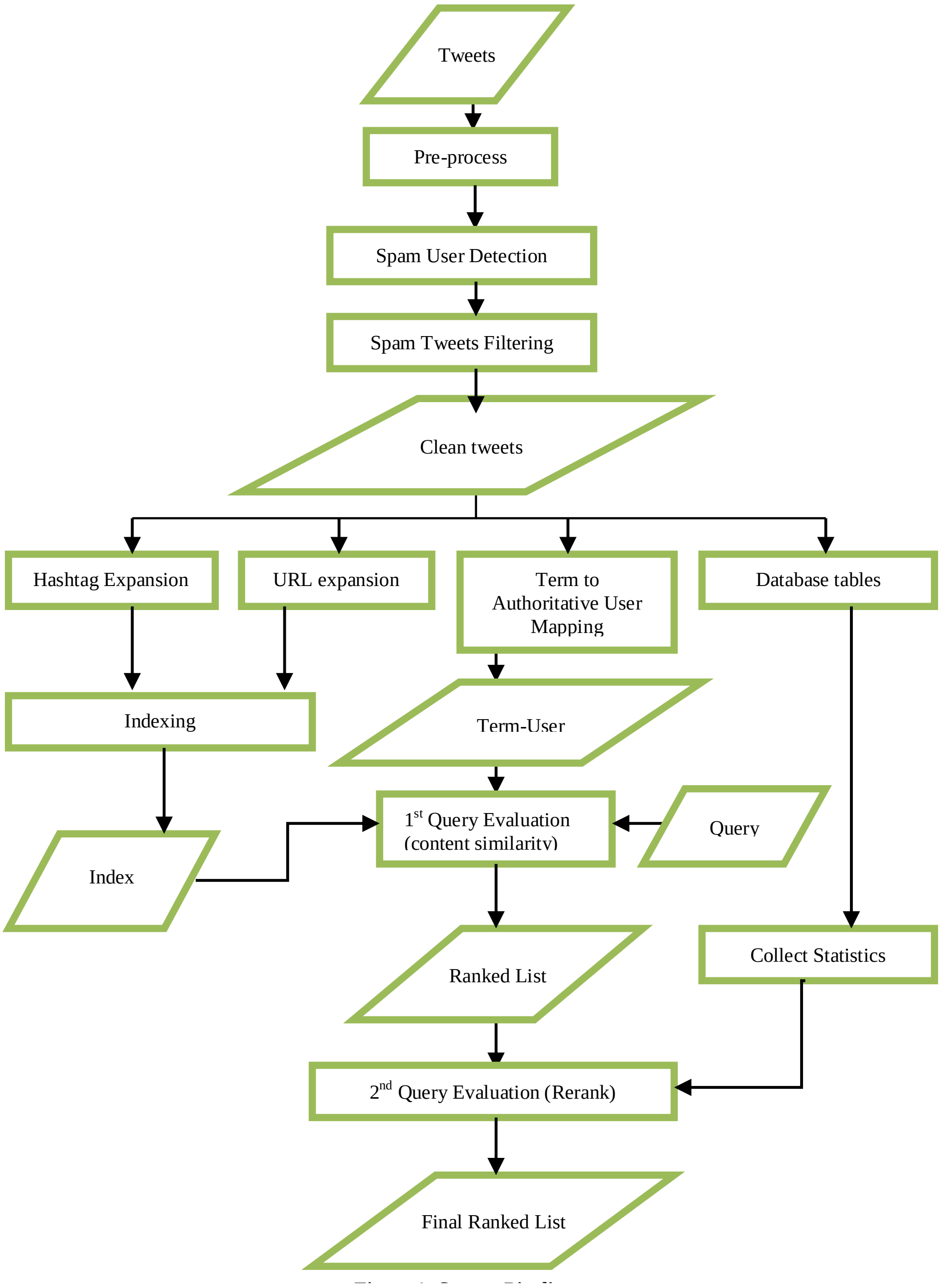

Figure 1: System pipeline showing an overview of the main blocks

the given collection of tweets, thus we decided to estimate it based on the number of replies to the user, number of users mentioning him/her, and number of users who retweeted any of his/her tweets. URL popularity is defined as a frequency of a URL appearance in the collection. The tweet popularity is defined as the frequency of retweeting it. Finally, user authoritative score is computed over the frequency of terms appearing in the user profiles. The final score is computed as a weighted sum over these scores. At the end of this stage, we get 50 tweets with the highest score at the re-ranking stage. The final step is to sort these 50 tweets by timestamp in descending order according to TREC requirement. The entire processing pipeline is illustrated in Figure 1.

4. EXPERIMENTAL EVALUATION

We conducted several experiments on our implementation of the BEST system. We focus here on the experiments related to the design of our tweet expansion and re-ranking feature. Both parts have several capabilities that we can enable and disable for our experiments and several parameters that were experimentally tuned. The results of the experiments are interesting because they helped us identify which of the features we considered were effective. The evaluation measure used in all of the following experiments is average precision at 30.

As we described earlier, we expand tweets with short term representation of hashtags and URLs with the goal of enriching the tweet content with terms that can describe the hashtag and URLs more verbosely. Re-ranking uses a combination of 5 features, as follows:

- content similarity
- user authority
- user popularity
- tweet popularity
- URL popularity

Our system combines these factors to recompute tweet scores after performing primary ranking, which considers only content similarity and recency.

The four combinations of enabling/disabling tweet expansion and re-ranking represent our four official submitted runs. Table 1 reports average precision at 30 (considering all relevant tweets) for each combination, while Table 2 only incorporates high relevant tweets.

Table 1: All-relevant evaluation (official runs)

	With Expansion	W/O Expansion
With Rerank	0.3354	0.3456
Without Rerank	0.3224	0.3347

Table 1 and Table 2 indicate that the re-ranking clearly improves the effectiveness, while expansion actually was harmful. Further analysis shows that expansion do enrich the content of a tweet, but this enrichment could be misleading.

Table 2: High-relevant evaluation (official runs)

	With Expansion	W/O Expansion
With Rerank	0.1202	0.1273
Without Rerank	0.1111	0.1162

Table 3: All-relevant by-score evaluation (official runs)

	With Expansion	W/O Expansion
With Rerank	0.3463	0.3571
Without Rerank	0.3224	0.3347

Table 3 shows the evaluation results by tweet reranking score instead of timestamp in descending order. It shows that there is a slight improvement over the the previous results which further strengths our observation that the re-ranking mechanism is effective.

After submitting the official runs, we did some modifications and fixed some bugs in our system and introduced five more additional "unofficial" runs:

1. Using only content similarity without spam user filtering, hashtag expansion, URL expansion and re-ranking
2. With spam user filtering but without any expansion and re-ranking
3. With hashtag expansion but without spam user filtering, URL expansion and re-ranking
4. With URL expansion but without spam user filtering, hashtag expansion and re-ranking
5. With re-ranking but without spam user filtering and any expansion.

Table 4 reports average precision at 30 (considering all relevant tweets) for these five runs while Table 5 only incorporates high relevant tweets.

Table 4: all-relevant evaluation (unofficial runs)

Without any feature	0.3707483
With Spam User Filtering	0.3707483
With Hashtag Expansion	0.3537415
With URL Expansion	0.37142858
With re-ranking	0.39455783

The scores above clearly indicate that the unofficial runs outperform the official ones after the modifications and bug fixes. We also notice that spam user filtering does not harm the precision at all, hashtag expansion was harmful, and URL expansion actually improves the results a bit. In our official submitted runs, we did not separate these two expansions so that we cannot tell which one is deleterious. Finally, as our official runs, the re-ranking feature effectively improves the system performance in Table 4. Surprisingly, there is a slight drop in precision comparing re-ranking run with the run without any feature according to Table 5; we need to do more investigation into that since Table 4 shows re-ranking feature effectively improves the precision.

Table 5: High-relevant evaluation (unofficial runs)

Without any feature	0.14343435
With Spam User Filtering	0.14343435
With Hashtag Expansion	0.13434343
With URL Expansion	0.14444445
With re-ranking	0.14141414

5. CONCLUSION

In this work, we present a first implementation of a search system based on Hadoop simulating real-time search in tweet stream. The main features we tried in that implementation include spam user detection, hashtag and URL expansion, estimation of user popularity and topic authority, and re-ranking. According to our initial experimental results and analysis, we found the spam user detection and re-ranking the most useful features that clearly improve efficiency and effectiveness respectively. However, we noticed that expansion techniques (in the simple forms we attempted) were not that beneficial and even sometimes harm the performance. We leave exploring more sophisticated ways of expansion and performing more experiments on leveraging combination of features to future work.

6. REFERENCES

[1] Akshay Java, Xiaodan Song, Tim Finin, Belle Tseng, 2007, *Why we twitter: understanding microblogging usage and communities.* In Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis. (WebKDD/SNA-KDD '07). ACM, New York, NY, USA, 56-65.

[2] Zhao, D. and Rosson, M.*How and why people Twitter: the role that micro-blogging plays in informal communication at work.* In Proceedings of the ACM 2009 international conference on supporting group work (GROUP '09). ACM, New York, NY, USA, 243-252

[3] Jaime Teevan, Daniel Ramage, and Merredith Ringel Morris. 2011. *TwitterSearch: a comparison of microblog search and web search.* In Proceedings of the fourth ACM international conference on Web search and data mining (WSDM '11). ACM, New York, NY, USA, 35-44.

[4] McFedries, P. 2007. *Technically speaking: All a-twitter.* IEEE Spectrum, 44(10), 84.

[5] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. *What is Twitter, a social network or a news media?.* In Proceedings of the 19th international conference on World wide web (WWW '10). ACM, New York, NY, USA, 591-600.

[6] Chun Chen, Feng Li, Beng Chin Ooi and Sai Wu. 2011. *TI: An Efficient Indexing Mechanism for Real-Time Search on Tweets.* In Proceedings of the 2011 international conference on Management of data (2011), pp. 649-660.

[7] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010.*TwitterRank: finding topic-sensitive influential twitterers.* In Proceedings of the third ACM international conference on Web search and data mining (WSDM '10). ACM, New York, NY, USA, 261-270.

[8] Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. 2010.*An empirical study on learning to rank of tweets.* In Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 295-303.

[9] Anlei Dong, Ruiqiang Zhang, Pranam Kolari, Jing Bai, Fernando Diaz, Yi Chang, Zhaohui Zheng, and Hongyuan Zha. 2010.*Time is of the essence: improving recency ranking using Twitter data.* In Proceedings of the 19th international conference on World wide web (WWW '10). ACM, New York, NY, USA, 331-340.

[10] William B. Cavnar and John M. Trenkle. 1994.*N-Gram-Based Text Categorization.*

[11] Tom White. 2009. *Hadoop: The Definitive Guide (1st ed.).* O'Reilly Media, Inc.

[12] Jeffrey Dean and Sanjay Ghemawat. 2004. MapReduce: simplified data processing on large clusters. In Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6 (OSDI'04), Vol. 6. USENIX Association, Berkeley, CA, USA, 10-10.